

Títol: Análisis y Diseño siguiendo RUP para los Terminales de Autoservicio del SOC

Volum: 1/2 - Memoria del Proyecto

Alumne: Miguel Ángel Marina Ramírez

Director/Ponent: Albert Botella Plana

Departament: Llenguatges i Sistemes Informàtics

Data: 12/06/2009

GESTIÓ ACADÈMICA-FIB
ENTRADA
26 JUN. 2009



Índice

1	Introducción	9
1.1	Motivación y Origen del Proyecto	9
1.2	El Servei d'Ocupació de Catalunya	9
1.3	Entorno	10
1.4	RUP	11
1.5	Oportunidad de negocio	13
1.6	Definición del Problema	13
1.7	Objetivos del Proyecto	14
1.8	Anexos asociados	15
2	Guía del Proyecto	17
2.1	Development Case	17
2.1.1	Introducción	17
2.1.2	Ciclo de vida	17
2.1.3	Roles	19
2.1.4	Guía de las diferentes disciplinas	19
2.1.5	Justificación artefactos	21
3	Análisis de Requisitos	25
3.1	Requisitos Funcionales	25
3.2	Requisitos no Funcionales	26
3.2.1	Requisitos de Look & Feel	27
3.2.2	Requisitos de Usabilidad y Accesibilidad	27
3.2.3	Requisitos de Rendimiento, Precisión, Escalabilidad y Mantenibilidad	28
3.2.4	Requisitos de Seguridad y Fiabilidad	29
3.2.5	Requisitos Legales	29
3.3	Mejoras Añadidas	29
3.4	Anexos asociados	30
4	Business Model	31
4.1	Introducción	31
4.2	Business Actors	32
4.3	Business Use Cases	33
4.4	Business Object Model	37
4.4.1	Entities y Workers del Business System de Administración	38
4.4.2	Entities y Workers del Business System de Formación	38
4.4.3	Entities y Workers del Business System de Intermediación	39
4.4.4	Entities y Workers del Business System de Información	40
4.4.5	Entities y Workers del Business System de Soporte	41
4.5	Business Use Case Realitization	42
5	Use Case Model	46



5.1	Introducción	46
5.2	Actores	46
5.3	Diagramas de Caso de Uso	47
5.3.1	Casos de Uso del Actor Ciudadano	49
5.3.2	Casos de Uso del Actor Ciudadano Inscrito	51
5.3.3	Casos de Uso del Actor Ciudadano Registrado	55
5.3.4	Casos de Uso del Actor Trabajador SOC	55
5.4	Descripción de las Realizaciones	56
5.5	Diagramas Actor-Sistema	58
6	Storyboards	62
7	Analysis Model	65
7.1	Diagrama de Clases	65
7.2	Realización de los Casos de Uso	68
7.2.1	Diagramas VOPC	69
7.2.2	Diagramas de Secuencia	72
8	Design Model	78
8.1	Patrones de Diseño	78
8.1.1	Patrón Iterador	78
8.1.2	Patrón Controlador	79
8.1.3	Patrón Experto y Creador	80
8.1.4	Otros patrones de diseño	81
8.2	Persistencia	81
8.3	Diagramas VOPC	83
8.4	Diagramas de Secuencia	84
9	Tecnologías y Arquitectura del Sistema	95
9.1	Descripción del Sistema	95
9.1.1	Base de Datos del Sistema	96
9.1.2	Sistema Operativo del Servidor	96
9.1.3	Servidor Web	97
9.1.4	Servidor de Aplicaciones	97
9.1.5	Datamapper	98
9.1.6	Framework	98
9.1.7	Lenguajes de Programación	99
9.2	Vista Lógica y Física	99
9.2.1	Descripción Física de los Servidores	99
9.2.2	Comunicación entre los Servidores (Capas)	100
9.3	Anexos asociados	101
10	Base de Datos	102
10.1	Diagrama Físico de Base de Datos	102
10.2	Optimización de Consultas: Estudio de Índices	106
10.3	Integridad del Modelo de Datos	110
10.3.1	Trigger realización Eliminar Cuenta	110
10.3.2	Triggers realización Actualizar Perfil	110
10.4	Implementación Física de la Base de Datos	111
11	Planificación y Gestión del Proyecto	112



11.1	Planificación Temporal	113
11.2	Coste Económico	117
12	Conclusiones	121
13	Bibliografía	124

Índice de Figuras

Figura 1: RUP Desarrollo en espiral.....	12
Figura 2: Fases RUP	12
Figura 3: Funcionalidades de los Puntos de Autoservicio.....	25
Figura 4: Modelado de Negocio	31
Figura 5. Business Actors.....	33
Figura 6. Business Use Case Gestión Cuentas	35
Figura 7. Business Use Case Gestión Información.....	35
Figura 8. Business Use Case Gestión Cursos	36
Figura 9. Business Use Case Gestión Ofertas.....	36
Figura 10. Business Use Case Soporte	37
Figura 11. Entities y workers Administración.....	38
Figura 12. Entities y workers Formación.....	39
Figura 13. Entities y workers de Intermediación.....	40
Figura 14. Entites y workers de Información.....	41
Figura 15. Entities y workers de Soporte.....	42
Figura 16. Business Use Case Realitization Inscripción Oferta	44
Figura 17. Business Use Case Realitization Rellenar Solicitud.....	45
Figura 18. Actores del Sistema	47
Figura 19. Caso de Uso Ciudadano-Gestión Cuenta y Demanda.....	49
Figura 20. Caso de Uso Ciudadano-Gestión Información.....	49
Figura 21. Caso de Uso Ciudadano-Gestión Formación.....	50
Figura 22. Caso de Uso Ciudadano-Gestión Ofertas.....	50
Figura 23. Caso de Uso Ciudadano-Soporte	51
Figura 24. Caso de Uso Ciudadano Inscrito-Gestión Cuenta y Demanda	51
Figura 25. Caso de Uso Ciudadano Inscrito-Gestión Información	52
Figura 26. Caso de Uso Ciudadano Inscrito-Gestión Formación	53
Figura 27. Caso de Uso Ciudadano Inscrito-Gestión Ofertas	54
Figura 28. Caso de Uso Ciudadano Inscrito-Gestión Sesión.....	55
Figura 29. Caso de Uso Ciudadano Registrado-Gestión Ofertas	55
Figura 30. Caso de Uso Trabajador SOC-GEstion Punto Autoservicio.....	56
Figura 31. Diagrama Actor-Sistema Buscar Cursos.....	60
Figura 32. Diagrama Actor-Sistema Crear Cuenta	61
Figura 33 Diagrama de Clases	66
Figura 34. VOPC Eliminar Cuenta.....	70
Figura 35. VOPC Buscar Ofertas	71
Figura 36. VOPC Logout	72
Figura 37. Analysis Model – Diagrama Secuencia Acceso Información	73
Figura 38. Diagrama Secuencia Análisis – Crear Cuenta.....	74
Figura 39. Diagramas Secuencia Análisis – Relaciones de Inclusión	74
Figura 40. Diagrama Secuencia Análisis – Actualizar Informes	75
Figura 41. Diagrama Secuencia Análisis – Informe DARDO.....	75
Figura 42. Diagrama de Secuencia Análisis - Wizard	76
Figura 43. Diagrama Secuencia Análisis – Crear Cuenta.....	76
Figura 44. Diagrama Secuencia Análisis – Barra de Herramientas	77
Figura 45. Design Model – Patrón Iterador	79
Figura 46. Design Model – Patrón Controlador Caso de Uso	80
Figura 47. Design Model – Funciones get y close de Hibernate	82
Figura 48. Design Model – Funciones flush y close de Hibernate.....	82
Figura 49. Design Model VOPC – Gestión Ofertas.....	84
Figura 50. Design Model – Diagrama Secuencia Acceso Información	87
Figura 51. Design Model – Boundary Mensaje Error	87
Figura 52. Design Model – Clases Session Factory y Session.....	87
Figura 53. Design Model. Funcion get Hibernate	88



<i>Figura 54. Design Model – Inscripción Abreviada.....</i>	<i>88</i>
<i>Figura 55. Design Model – Crear Cuenta</i>	<i>89</i>
<i>Figura 56. Design Model – Eliminar Cuenta.....</i>	<i>90</i>
<i>Figura 57. Design Model – Inscripción Abreviada Cursos Recomendados.....</i>	<i>91</i>
<i>Figura 58. Design Model – Bloqueo de una cuenta.....</i>	<i>92</i>
<i>Figura 59. Design Model – Clase Soluciones Soporte.....</i>	<i>93</i>
<i>Figura 60. Design Model – Búsqueda soluciones soporte</i>	<i>93</i>
<i>Figura 61. Design Model – Notificar inscripciones a empresas</i>	<i>94</i>
<i>Figura 62. Resumen arquitectura del Sistema</i>	<i>95</i>
<i>Figura 65. Diagrama físico de Base de Datos</i>	<i>105</i>

1 Introducción

1.1 Motivación y Origen del Proyecto

La motivación personal que determinó la realización de este Proyecto de Final de Carrera tiene su origen en la asignatura de *Projecte d'Enginyeria del Software i Bases de Dades*. Esta asignatura me pareció realmente interesante y enriquecedora, ya que permite poner en práctica gran parte de los conocimientos adquiridos a lo largo de las asignaturas de Ingeniería del Software cursadas durante la carrera. No obstante, la lógica limitación temporal de la asignatura impidió profundizar y explorar una serie de aspectos y tareas asociados al proceso de análisis y diseño de un Sistema de Información.

Así pues, la realización del Proyecto de Final de Carrera se presentó como una oportunidad inmejorable de ampliar y profundizar en el proceso de especificación y diseño de un Sistema. Gracias a la colaboración de Albert Botella, llegó a mis manos un **documento de preinscripciones técnicas** de carácter público elaborado por el **Servei d'Ocupació de Catalunya (SOC)** que describe minuciosamente las características de un Sistema de Información basado en una serie de **Puntos de Autoservicio**. Este Sistema responde a la necesidad del SOC de establecer una nueva vía de comunicación con los Ciudadanos a través de un conjunto de Terminales de Autoservicio. Así pues, este documento definido por el propio SOC establecía las bases para que diversas empresas presentaran sus propuestas de diseño e implementación de un Sistema que cumpliera con los requisitos especificados en dicho documento.

Por tanto, este documento de preinscripciones técnicas, que corresponde obviamente a un **proyecto real**, se convirtió en el punto de inicio de este Proyecto de Final de Carrera, ya que partiendo de este documento, se ha procedido a realizar todo el análisis y diseño del Sistema, profundizando y adentrándose en una serie de tareas asociadas a la arquitectura e implementación física de dicho Sistema.

1.2 El Servei d'Ocupació de Catalunya

El *Servei d'Ocupació de Catalunya (SOC)* es un organismo autónomo de carácter administrativo, adscrito al Departamento de Trabajo. El SOC enmarca su actuación en la Estrategia Europea sobre Trabajo y en los planes nacionales sobre trabajo del Estado.

El SOC tiene como funciones principales:

- Ofrecer y dar servicio a todas las personas que buscan trabajo, independientemente de la situación laboral que presente.
- Ofrecer un marco de igualdad de oportunidades laborales para todos los ciudadanos.
- Fomentar el espíritu emprendedor y dar soporte a la pequeña y mediana empresa.

- Conseguir que el diálogo y compromiso entre los actores públicos y privados sean el motor de las políticas de ocupación.
- Conseguir un nivel de ocupación alto mediante una política de acción activa.

El SOC presenta dos niveles de actuación:

1. Orgánico: El Consejo de Dirección es un órgano de gobierno.
2. Oficinas de Trabajo:
 - Los Servicios Territoriales, que son la representación institucional del SOC en un territorio concreto.
 - Las oficinas de Trabajo de la Generalitat de Catalunya, que son el punto de referencia en un territorio.
 - Los Centros de Innovación y Formación Ocupacional (CIFO), que son los centros de referencia de formación profesional.

Así pues, el *Servei d'Ocupació de Catalunya* es el organismo que se ocupa de la gestión e integración del conjunto de actuaciones y servicios ocupacionales destinados a mantener y fomentar la ocupación estable y de calidad. Entre los objetivos principales de su actuación, cabe destacar:

- Prevenir y reducir la desocupación.
- Ofrecer protección a las personas desocupadas fomentando y garantizando la Formación Profesional Ocupacional y continuada.
- Ejercer la intermediación en el mercado laboral.
- Garantizar la orientación y la readaptación profesional para conseguir mejoras en la ocupación.

Para culminar con éxito estos objetivos marcados, el SOC dispone de la siguiente organización departamental:

- Dirección del SOC.
 - Dirección Red Ocupacional:
 - Dirección Territorial Barcelona.
 - Dirección Territorial Tarragona.
 - Dirección Territorial Lleida.
 - Dirección Territorial Girona.
 - Dirección Territorial Catalunya Central.
 - Dirección Programas:
 - Subdirección General de Políticas Activas.
 - Secretaría Técnica.
 - Gerencia.

1.3 Entorno

Un aspecto importante del entorno que es necesario remarcar es que tanto el *Servei d'Ocupació de Catalunya* como otros organismos públicos y privados de

carácter administrativo, están experimentando una serie de cambios tecnológicos debidos a los avances técnicos y científicos del mundo globalizado en que nos encontramos, así como el creciente y rápido acceso y adaptación de la población a las nuevas tecnologías. Estos factores están llevando a estos organismos administrativos a apostar por soluciones tecnológicas innovadoras con el fin de automatizar dichos procesos administrativos.

1.4 RUP

El **Rational Unified Process (RUP)** es una metodología orientada al proceso de desarrollo de sistemas Software que junto con el lenguaje de especificación UML constituyen la herramienta más utilizada para el análisis, diseño, implementación y documentación de Sistemas Orientados a Objetos.

Realmente, no se puede afirmar que el RUP sea un sistema con pasos estrictamente establecidos, sino un conjunto de metodologías adaptables a las necesidades del proyecto a desarrollar.

Esta metodología se basa en cinco principios fundamentales:

- **Adaptación del proceso:** El proceso de desarrollo debe adaptarse a las características tanto del Sistema a desarrollar como de la organización que debe desarrollarlo.
- **Equilibrio de prioridades:** Debe encontrarse un equilibrio entre las necesidades y deseos de todos los participantes del proyecto, participantes que habitualmente suelen tener requerimientos contradictorios o contrapuestos.
- **Demostración de valor de forma iterativa:** A lo largo del desarrollo del proyecto, se deben producir entregas iterativas del contenido del mismo aunque estas entregas se produzcan únicamente a nivel interno. Estas entregas permitirán constatar la opinión de los inversores, así como evaluar la calidad y estabilidad del producto permitiendo de este modo corregir posibles desviaciones o errores.
- **Elevar el nivel de abstracción:** Este factor debe suponer una motivación para la utilización de elementos reutilizables tales como patrones de diseño o *Frameworks*, evitando que los ingenieros de software pasen directamente los requisitos a la codificación de un sistema software a la medida del cliente.
- **Control de calidad presente durante toda la producción:** El control de calidad del producto no debe realizarse únicamente al finalizar una iteración, sino que debe estar presente durante todo el proceso de producción.

El ciclo de vida de una metodología como RUP es el encargado de organizar las tareas en fases e iteraciones. En la metodología RUP, el ciclo de vida es una implementación del Desarrollo en Espiral, es decir, las actividades a realizar conforman una espiral en la que cada bucle o iteración representa un conjunto de ellas. Las actividades que se realizan en cada iteración dependen del resultado de la anterior.

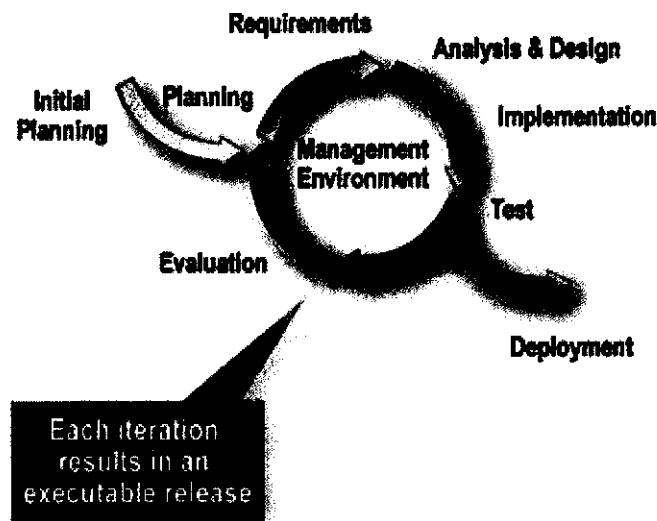


Figura 1: RUP Desarrollo en espiral

RUP divide el proceso de desarrollo en cuatro fases:

- **Inicio**, centrada en actividades como el Modelado de Negocio y el Análisis de Requisitos del Sistema.
- **Elaboración**, centrada en actividades como el análisis y diseño de las funcionalidades del Sistema.
- **Construcción**, centrada en la construcción física del Sistema.
- **Transición**, centrada en garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

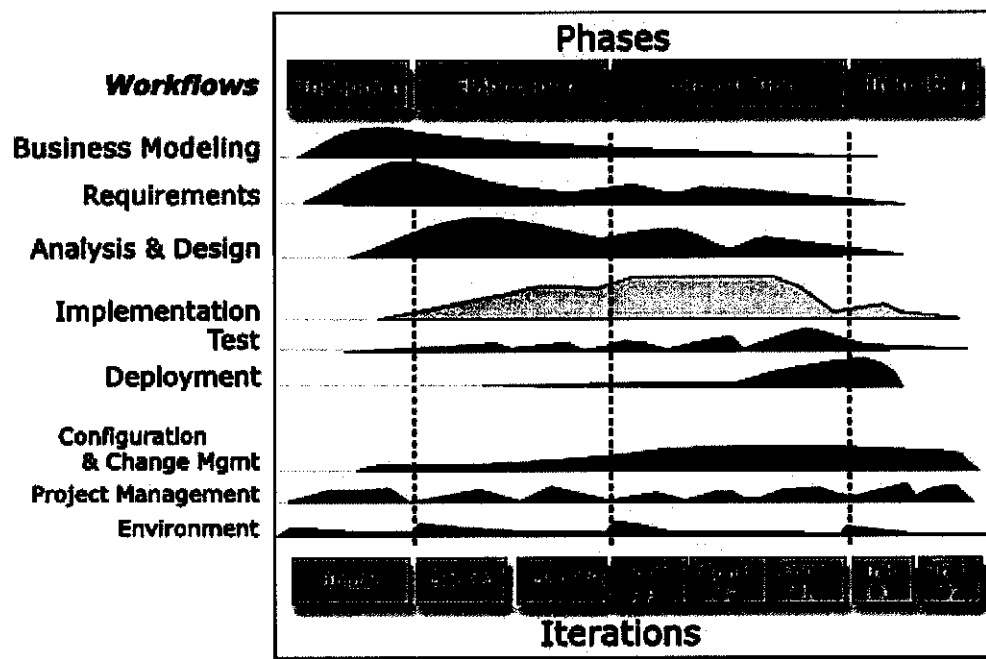


Figura 2: Fases RUP

Dado que, como se ha comentado anteriormente, uno de los principios de la metodología RUP es la **Adaptación de Proceso**, se ha definido una adaptación o instanciación de la metodología RUP para el desarrollo del Sistema de Puntos de Autoservicio del SOC, ajustando la aplicación de dicha metodología a las características del proyecto en cuestión. Esta adaptación puede consultarse en el apartado "Guía de Proyecto".

1.5 Oportunidad de negocio

Como se ha comentado previamente, el documento de preinscripciones técnicas definido por el SOC abre la veda para la presentación de propuestas de diseño e implementación del Sistema de Puntos de Autoservicio por parte de diferentes empresas privadas. De entre todas estas propuestas, el SOC seleccionará la más apropiada en base a una serie de premisas definidas en el propio documento de preinscripciones. Así pues, de manera paralela a este Proyecto de Final de Carrera, el Sistema de Puntos de Autoservicio que se va a desarrollar proporcionará a sus usuarios un nuevo canal de comunicación con el **Servei d'Ocupació de Catalunya** a través de una serie de cajeros automáticos distribuidos entre las distintas oficinas que el SOC tiene repartidas por Cataluña. Estos cajeros automáticos o Puntos de Autoservicio posibilitarán que un ciudadano realice los trámites y servicios administrativos que habitualmente realiza en las oficinas del SOC mediante la intervención de un funcionario utilizando un sistema automatizado y sin requerir la participación de un trabajador del SOC.

Esta nueva vía de comunicación pretende, por tanto, aportar agilidad y fluidez a la realización de dichas gestiones y trámites por parte del ciudadano, reduciendo de esta manera tanto el personal del SOC destinado actualmente a gestionar estos servicios administrativos como el tiempo que un ciudadano debe destinar a este tipo de actividades. Los servicios administrativos que podrán realizarse a través de estos cajeros automáticos abarcan múltiples contextos que van desde consultar la información que el ciudadano debe entregar al SOC para realizar un trámite concreto hasta la búsqueda de ofertas e inscripción en cursos de formación destinados a cumplir los requisitos requeridos en dicha oferta.

1.6 Definición del Problema

Se pretende desarrollar un sistema de información que permita a los ciudadanos realizar de manera ágil y eficaz los trámites burocráticos y servicios administrativos ofrecidos y gestionados por el SOC. El SOC necesita disponer de algún medio que permita a los ciudadanos consultar la información requerida para realizar un trámite o servicio administrativo concreto, gestionar la búsqueda e inscripción de ofertas de empleo por parte de los ciudadanos, así como la búsqueda e inscripción de cursos de formación destinados a cumplir los requisitos demandados en una oferta concreta.

Se necesita por tanto almacenar de manera persistente tanto la información de los ciudadanos que realizan estos trámites y gestiones como la información asociada a las ofertas de empleo y cursos de formación ofertados y destinados a dichos ciudadanos. Del mismo modo, se deberán registrar las inscripciones que realicen los ciudadanos con las ofertas y cursos ofrecidos.

El SOC pretende también que los ciudadanos, a través de estos puntos de Autoservicio instalados en sus oficinas, puedan consultar información sobre los servicios ofrecidos por el SOC y las distintas prestaciones destinadas a estos ciudadanos.

Por otra parte, el sistema de información que se pretende desarrollar debe permitir a los usuarios del mismo rellenar y por supuesto entregar una serie de solicitudes y formularios encaminados a gestionar y percibir los servicios y prestaciones que el SOC destina a los ciudadanos. Además de que el usuario pueda entregar información al SOC en forma de formularios y solicitudes mediante estos puntos de Autoservicio, el sistema deberá del mismo modo permitir al ciudadano solicitar una serie de informaciones relacionados con las prestaciones que percibe, servicios a que está inscrito o sobre sus datos personales

Además estos puntos de Autoservicio deben permitir al ciudadano actualizar cuando lo desee su información personal y perfil que el SOC guarda sobre su persona, así como renovar periódicamente sus demandas de empleo.

1.7 Objetivos del Proyecto

El Proyecto de Final de Carrera “**Análisis y Diseño siguiendo RUP para los Terminales de Autoservicio del SOC**” pretende establecer una guía y una base sólida que sustente y facilite la implementación de dichos terminales. Este proyecto se inicia partiendo de una documentación técnica de carácter público definida por el *Servei d'Ocupació de Catalunya* en la cual se detallan las características y requisitos que deben cumplir los Terminales o Puntos de Autoservicio destinados a agilizar una serie de procesos administrativos.

Partiendo de esta documentación asociada lógicamente a un proyecto real, se pretende realizar un análisis y diseño exhaustivo de cada una de las funcionalidades ofrecidas en dichos terminales, estudiando minuciosamente cada decisión con el fin de que la posterior construcción e implementación del Sistema se limite únicamente a plasmar esta especificación y diseño en un lenguaje de programación ya definido en dicho análisis.

No obstante, dada la naturaleza de un Proyecto de Final de Carrera, se pretende que el proyecto supere la frontera del análisis y diseño establecida por el RUP y se adentre en la implementación y construcción del Sistema. Para ello, se ha procedido a construir físicamente en un SGBD concreto la Base de Datos del Sistema y el diseño de la interficie gráfica o *StoryBoards* no se ha limitado a un simple diseño sobre papel, sino que se ha realizado la implementación física de dicha interficie gráfica permitiendo la interacción y navegación por las diferentes pantallas del Sistema.

Así pues, para poder realizar esta serie de propósitos, se ha procedido a llevar a cabo las siguientes tareas:

- Instanciar el RUP para determinar que artefactos de dicha metodología se ajustan a las características del proyecto que nos concierne.

- Evaluación del modelo de negocio del SOC con el fin de esclarecer las necesidades del Sistema que se pretende desarrollar.
- Definición de los requisitos funcionales y no funcionales del Sistema, prestando especial atención a estos últimos en base a las premisas definidas por el cliente en el documento de especificaciones técnicas.
- Realizar una planificación y gestión minuciosa de los recursos del proyecto evaluando los posibles riesgos que puedan acontecer y definiendo un plan de mitigación y contingencia para dichos riesgos. Esta planificación no abarca únicamente las etapas de análisis y diseño asociadas al proyecto que nos concierne, sino que se extiende hasta la construcción y puesta en marcha del Sistema.
- Especificación y análisis mediante UML de cada una de las funcionalidades del Sistema.
- Diseño de dichas funcionalidades mediante UML en base a una serie de patrones de diseño.
- Diseño e implementación de las pantallas que conforman la interficie gráfica de los terminales de Autoservicio.
- Diseño e implementación física de la Base de Datos del Sistema en el SGBD seleccionado incluyendo tareas de optimización de acceso a los datos almacenados y de mecanismos encargados de mantener la integridad del modelo.
- Especificación y Diseño de la arquitectura Sistema, así como una descripción del despliegue de la misma.
- Definición de un exhaustivo y detallado plan de pruebas que permita verificar el correcto funcionamiento del Sistema una vez realizada la implementación física del mismo.

1.8 Anexos asociados

Cabe destacar que esta documentación está formada por 2 volúmenes. El segundo volumen contiene una serie de anexos que amplían el contenido incluido en este primer volumen. A lo largo de los diferentes apartados que conforman este primer volumen se harán diversas referencias a estos anexos. Así pues, a continuación se muestra una lista de los anexos que componen el segundo volumen de esta documentación.

1. *Presinscripciones Técnicas de los Puntos de Autoservicios:* Este anexo contiene el documento original elaborado por el SOC en el cual se describen los requisitos que debe cumplir el Sistema de Puntos de Autoservicios

2. **Visión:** Anexo en el que se recogen y definen las necesidades del problema a solucionar, así como las características de la solución propuesta para dicho problema.
3. **Supplementary Specifications:** Este anexo amplía la descripción de los requisitos no funcionales del Sistema.
4. **Interfaz de Usuario:** En este anexo se describe de manera detallada el diseño de la Interfaz de Usuario, realizando una detallada comparación entre dicho diseño y los requisitos no funcionales definidos previamente.
5. **Software Architecture Document:** Este anexo recoge una visión global de la arquitectura que conforma los puntos de Autoservicio, así como una descripción y justificación de cada una de las decisiones tomadas con respecto al diseño de dicha arquitectura.
6. **Descripción de las Realizaciones de los Casos de Uso:** Anexo en el cual se recoge la descripción textual de las realizaciones que conforman los Casos de Uso del Sistema.
7. **Scripts de Base de Datos:** Anexo en el cual se incluye el código SQL y PL/SQL asociado a la creación de la Base de Datos del Sistema.
8. **Contratos de las Operaciones:** En este anexo se incluyen los contratos de cada una de las operaciones del Sistema.
9. **Pruebas del Sistema:** Anexo en el cual se describen de manera detallada tanto el *Test Plan* como los diferentes *Test Cases* definidos.
10. **Implementation Model:** En este anexo se describen de manera detallada cada uno de los componentes que conforman la Arquitectura del Sistema.
11. **Deploy Model:** Anexo en el cual se determina la distribución de los componentes y procesos que conforman el Sistema en la infraestructura de máquinas y redes del mismo.
12. **Software Development Plan.** En este anexo se muestra de manera detallada el proceso de planificación y gestión de los recursos del proyecto, incluyendo además un detallado análisis de riesgos.
13. **Glosario:** Anexo en el cual se incluye la definición de una serie de conceptos que aparecen a lo largo de esta documentación con el fin de facilitar la comprensión de la misma.

Así pues, y por lo que respecta a este primer apartado de Introducción, en el anexo "*Preinscripciones Técnicas de los Puntos de Autoservicio*" adjuntado en esta memoria se pueden consultar los detalles del documento de especificaciones técnicas definido por el *Servei d'Ocupació de Catalunya*, en el cual se detallan las características y requisitos que deben cumplir los Puntos de Autoservicio que este proyecto pretende diseñar.

Por otra parte, el anexo "*Visión*" contiene, entre otras cosas, una descripción problema que se pretende abordar, así como una descripción de los *Stakeholders* afectados por el desarrollo del este Sistema de Puntos de Autoservicio. El contenido de este anexo puede considerarse una ampliación a la información descrita en este apartado de Introducción.

2 Guía del Proyecto

Este apartado es una guía que describe la adaptación de la metodología RUP al proyecto que nos concierne. Esta adaptación se describe en los siguientes apartados:

2.1 *Development Case*

2.1.1 Introducción

2.1.1.1 Propósito

Este apartado tiene como objetivo describir que disciplinas y artefactos de la metodología RUP se utilizarán y con que formato. Además, se asignarán los diferentes artefactos a cada una de las fases en que se dividirá el proyecto.

2.1.1.2 Alcance

El alcance de este documento es principalmente para las fases de Inicio y Elaboración. No obstante, incluye una referencia a las primeras actividades de la fase de Construcción

2.1.1.3 Referencias

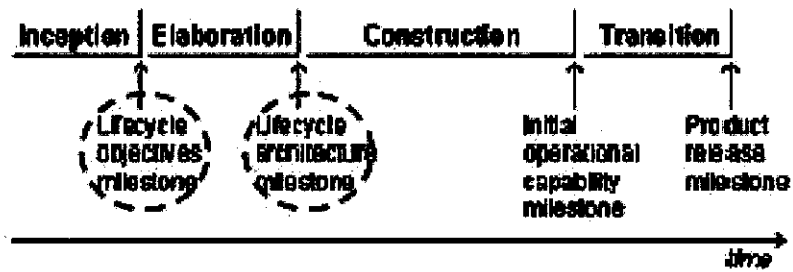
- Rational Unified Process 2003

2.1.1.4 Resumen

Se trata de proporcionar una descripción general del ciclo de vida en fases, con los objetivos y artefactos más significativos de cada una de ellas.

2.1.2 Ciclo de vida

El proyecto a desarrollar abarca principalmente las fases de Inicio y Elaboración de un proyecto siguiendo el modelo del RUP. El objetivo es definir una arquitectura implementable y un plan detallado de construcción del sistema.



2.1.2.1 Configuración de disciplinas

Business Modelling				
Business Use Case Model	X			
Business Use Case Realization	X			
Business Object Model	X			
Requirements				
Vision	X	X		
Glossary	X	X		
Use Case Model	X	X		
Use Case Specifications	X	X		
Supplementary Specifications	X			
Use Case StoryBoards		X		
Analysis & Design				
Software Architecture Document	X	X	X	
Analysis Model	X	X		
Design Model		X	X	
Data Model		X	X	
Navigation Map			X	
Implementation Model			X	
Deployment Model			X	
Test				

Test Plan		X	X	
Test Case			X	
Project Management				
Software Development Plan	X			
Iteration Plan	X	X	X	
Implementation				
Data Model Implementation				X

2.1.3 Roles

En el *Software Development Plan* se especificará la organización del proyecto y las capacidades que tiene cada persona. Estas capacidades o habilidades determinarán las actividades que deberá llevar a cabo esta persona durante la realización del proyecto. De acuerdo con el *Development Case*, deben existir habilidades para poder ejecutar las siguientes disciplinas RUP:

- Modelado de Negocio (*Business Modeling*)
- Requisitos
- Análisis y Diseño
- Pruebas
- Implementación
- Gestión de Proyecto

2.1.4 Guía de las diferentes disciplinas

2.1.4.1 Business Modeling

Se realizará fundamentalmente para modelar los procesos de negocio mediante diagramas de actividad. Entendemos por proceso de negocio, todas aquellas actividades necesarias para poder ofrecer un servicio.

Durante la realización del *Business Modeling*, se identificarán actividades, *workers* y entidades de negocio. Los artefactos más importantes son *Business Use Case*, *Business Use Case Realization* y *Business Object Model* (*Business Actor*, *Business Worker* i *Business Entity*).

Se identificarán y modelarán:

- Los servicios que ofrece el negocio, los cuales modelaremos como *Business Use Cases*.
- Los entes externos a los cuales el Business ofrece estos servicios (clientes) y los entes con los que colabora el Business para ofrecerlos (proveedores). El papel o rol que ocupan en un *Business Use Case* lo modelaremos como *Business Actors*.
- Los trabajadores internos que permiten realizar el servicio descrito en el *Business Use Case*: *Business Workers*.

- Las entidades de información que manipulan los *Business Workers* para ofrecer un servicio: *Business Entities*.
- La forma en que colaboran los *Business Workers* entre si e interacción con los *Business Actors*. Esto se modelará mediante procesos de negocio, los cuales modelaremos mediante diagramas de actividad de UML: *Business Workflows*. Asociado a cada *Business Use Case* existen uno o varios procesos de negocio dentro del *Business*.

Los *Business Use Case* se describirán en alto nivel y el análisis del negocio se centrará sobretodo en como se realizan estos *Business Use Cases*, es decir, en los *Business Use Case Realization*, para eso se identificarán los *Business Workers* y las *Business Entities*.

Primero de todo, definiremos el *Business Use Case Model*, identificando los *Business Actors* (actores externos a la organización) y *Business Use Cases* (procesos vinculados a los *Business Actor*

Seguidamente, se procederá a definir el *Business Object Model*, identificando *Workers* y *Business Entities*.

Se realizarán los diagramas de actividad para cada *Business Use Case* con los que generaremos los *Business Use Case Realization*.

2.1.4.2 Requisitos

En el proyecto que se va abordar no se realizará una Gestión de Requisitos, pero si su obtención y especificación.

Una primera descripción textual de los requisitos se incluirá en el documento "*Vision*", que será la primera descripción global del proyecto. Los requisitos funcionales se convertirán en los casos de uso del sistema. El documento "*Supplementary Specifications*" contiene la una especificación textual más detallada de los requisitos no funcionales.

Por otra parte, se elaborarán *StoryBoards* para describir como se realizan los *Casos de Uso* a nivel de interacción entre el usuario y la interficie del sistema. Cabe destacar que esta elaboración de los *StoryBoards* no se limitará a un simple diseño gráfico de los mismos, sino que se realizará una implementación de estos *StoryBoards* que permita observar la interacción entre el usuario y el sistema.

2.1.4.3 Analysis & Design

Las tareas de análisis y diseño se centran en el análisis de los Casos de Uso del sistema, definiendo las operaciones que se realizan en el interior del sistema: las clases de objetos necesarias, como se relacionan entre si y como colaboran para responder a las interacciones con el usuario especificadas en el Caso de Uso.

Por otra parte, se definirá la arquitectura del sistema mediante las diferentes vistas (Casos de Uso, Modelo de Análisis, Modelo de Diseño, Modelo de Datos, Componentes, Modelo de Procesos y despliegue tanto de componentes como de procesos en diferentes máquinas/redes).

Además, se definirán también ventanas, formularios y menús que tendrá el sistema y como se navega entre ellos (*Navigation Map*).

2.1.4.4 Implementation

Por lo que respecta a la implementación del sistema, se una propuesta para la arquitectura inicial del sistema. Además, cabe destacar, que se realizará la implementación física de la Base de Datos del Sistema.

2.1.4.5 Test

Se definirá un *Test Plan* y diferentes *Test Cases* para los casos de uso que conforman el sistema.

2.1.4.6 Project Management

Las tareas asociadas a la gestión del proyecto que se llevarán a cabo se centran en la definición de un *Software Development Plan* para el proyecto completo. Al iniciar cada una de las fases del mismo, se deberá realizar un plan detallado para la misma.

Por otra parte, se realizará un estudio de la viabilidad económica del proyecto. Este estudio se incluirá en el *Software Development Plan*.

Por último, se realizará un análisis de riesgos con el fin de identificarlos y definir un plan para su gestión.

2.1.5 Justificación artefactos

Artefacto	Descripción y Justificación
Business Model	La realización del Business Model nos permitirá obtener una visión global del funcionamiento actual de la empresa. Esta descripción facilitará la comprensión del problema a tratar y de la solución que se debe aportar. Este modelo será la base que guiará la realización de los Casos de Uso del sistema.
Vision	Se trata de un documento imprescindible en la realización de cualquier proyecto, ya que contiene la primera definición del problema a tratar, así como los requisitos que debe cumplir la solución aportada o las diferentes personas que se verán afectadas por el sistema que se va a desarrollar. Además, se ha decidido incluir una gestión de los eventuales riesgos que pueden producirse durante la realización del proyecto con el fin de mitigarlos y establecer un plan de acción para resolverlos. Debido a la gran variedad y cantidad de riesgos que pueden producirse en un proyecto de estas características, se antoja imprescindible realizar esta gestión de riesgos.

Glossary	La documentación de este proyecto, como en la mayoría de los que se desarrollan actualmente, será accesible a personas que no conocen a fondo del desarrollo del mismo ni los detalles que lo definen. Por esta razón, se considera necesario definir un documento que recoja la definición de una serie de términos técnicos o propios del proyecto con el fin de asegurar la comprensión de la documentación por parte de estas personas.
Use Case Model	El Use Case Model se puede calificar como un artefacto básico e imprescindible para la realización de cualquier proyecto, ya que permite observar la secuencia de interacciones que realizan los usuarios con el sistema.
Supplementary Specifications	Como se comenta en el documento de especificaciones técnicas aportado por el cliente, una de las claves del éxito del proyecto será una definición adecuada y óptima de la interficie del sistema, ya que las características de usabilidad del mismo deben permitir que personas no habituadas a interaccionar con sistemas como automatizados no rechacen la utilización del sistema. Así pues, se ha decidido realizar un análisis profundo de los requisitos no funcionales del sistema con el fin de alcanzar este objetivo.
StoryBoards	Como ya se ha comentado anteriormente, el diseño la interficie del sistema es uno de los factores que determinaran el éxito del sistema. Así pues, es necesario realizar un análisis y diseño detallado de la interficie del sistema mediante la definición de los StoryBoards. Cabe destacar además, que la realización de este artefacto no se limitará a un simple diseño gráfico del mismo, sino que se realizará una implementación de dicho artefacto con el fin de mostrar la interacción del usuario con las distintas pantallas del sistema.
Software Architecture Document	Se trata de una especie de memoria técnica del proyecto, cuyo contenido esta formado por elementos de otros artefactos definidos previamente. Su realización viene justificada porque permite mostrar de visión global de la arquitectura del sistema, así como una descripción de las diferentes decisiones tomadas durante la realización del proyecto con respecto al diseño de dicha arquitectura.
Analysis Model & Design Model	Se trata de 2 modelos básicos y absolutamente imprescindibles para analizar y diseñar la arquitectura de un sistema de información como el que se pretende desarrollar. Estos 2 modelos permitirán definir la estructura interna del sistema, las operaciones necesarias para cada una de las funcionalidades del mismo, así como la interacción entre cada uno de los elementos que conforman dicha estructura interna.
Data Model	Al igual que en la mayoría por no decir totalidad de proyectos asociados a un sistema de información, el proyecto que estamos tratando está íntimamente ligado a una Base de Datos. Así pues, se antoja indispensable realizar un análisis y diseño pormenorizado de esta Base de Datos con el fin de analizar la fuente de datos del sistema que vamos a desarrollar.
Navigation Map	Este artefacto nos permitirá conocer la relación entre las

	diferentes pantallas que conforman la interficie del sistema. Como se ha destacado a lo largo de este documento, el éxito del proyecto esta muy arraigado a un diseño adecuado de dicha interficie. Así pues, una evaluación y descripción de la relación entre las diferentes pantallas de la interficie se antoja cuanto menos necesaria.
Deployment Model	Este artefacto se elaborará con el objetivo de establecer una descripción una descripción detallada de la distribución de cada unos de los componentes y procesos en la infraestructura de redes y máquinas. Se pretende determinar la separación lógica de los componentes del sistema en cada una de las capas que conforman la arquitectura. Se ha considerado necesaria la creación de este artefacto dado que se pretende que toda la documentación aportada en el proyecto no sea una simple base para su construcción, sino que permita que esta construcción del sistema se limite a la finalización la implementación física del mismo
Implementation Model	Este artefacto permitirá evaluar y posteriormente definir con detalle cada uno de los componentes que conforman la arquitectura del sistema. Este análisis permitirá guiar la implementación y construcción física del sistema, aportando como se ha comentado anteriormente algo más que una base para realizar dicha implementación.
Test Plan	Este artefacto contiene una descripción detallada del plan de test que se llevara a cabo en la construcción del proyecto. Este documento permitirá preveer posibles errores en la construcción del sistema facilitando en gran medida la tarea de los programadores de dicho sistema y garantizando una realización del mismo más rápida, segura y económica.
Test Case	Se trata de un artefacto que se deriva de las acciones programadas en el Test Plan. Consiste en una descripción detallada de las pruebas que se realizarán para cada uno de los Casos de Uso del sistema.
Software Development Plan	Considerando la envergadura del proyecto, se ha considerado necesaria realizar una gestión y planificación de las tareas que se llevarán a cabo y de los recursos que destinarán para alcanzar los objetivos fijados. Una descripción de la estructura organizativa del proyecto, la planificación de las diferentes fases del mismo (detalladas en el artefacto <i>Iteration Plan</i>) y una gestión de los posibles riesgos conformarán la información asociada a este artefacto.
Iteration Plan	Dadas las dimensiones del proyecto que estamos tratando, resulta imprescindible realizar una planificación detallada de las tareas que se llevarán a cabo en cada una de las fases del proyecto. Determinar estas tareas, la duración y recursos que se destinarán a su realización, así como las relaciones que se establecen entre ellas conformarán este artefacto. Con este artefacto, se pretende establecer un control y planificación de cada una de las tareas que se realizarán permitiendo de esta manera detectar y corregir con la mayor brevedad posible las desviaciones que puedan surgir.
Data Model	Del mismo modo que se ha hecho con otros artefactos, la



Implementation	implementación de la Base de Datos del sistema responde al objetivo marcado de dejar el sistema listo para que únicamente requiera que se finalice la implementación del código del mismo. Así pues, se llevará a cabo la implementación física de la Base de Datos diseñada en el <i>Data Model</i> , así como la inserción de una serie de datos en la misma.
-----------------------	---

3 Análisis de Requisitos

Este apartado contiene una especificación de los requisitos que definirán las características del Sistema. Para ello, se realizará un análisis de tanto los requisitos funcionales como no los funcionales.

Los requisitos funcionales describen el comportamiento esperado del Sistema. Cada uno de ellos expresa una relación entre las entradas y salidas del Sistema, es decir, especifica que salidas deben producirse a partir de unas determinadas entradas, así como las operaciones necesarias para obtener dichas salidas. Además, se debe especificar el comportamiento del Sistema ante situaciones anómalas tales como errores, entradas no válidas...

Por otra parte, el Sistema tiene que cumplir una serie de características ajenas a la funcionalidad del mismo. Estas características sirven para fijar criterios sobre el aspecto que debe tener el sistema definiendo factores como la usabilidad, el rendimiento, la seguridad, etc. Como se remarca en el documento de preinscripciones técnicas definido por el SOC, estos requisitos no funcionales cobran una gran importancia en el Sistema de Puntos de Autoservicio, dado que factores como la usabilidad resultan fundamentales en un Sistema automatizado destinado a sectores de la sociedad no excesivamente acostumbrados a interactuar con sistemas de similares características.

3.1 Requisitos Funcionales

Los puntos de Autoservicio instalados en las oficinas del SOC facilitarán una serie de funcionalidades destinadas a que los ciudadanos puedan tanto proporcionar información personal requerida por el SOC como obtener información de los servicios administrativos y trámites que debe realizar. Estas funcionalidades han sido descritas por el propio *Servei d'Ocupació de Catalunya* en el documento de preinscripciones técnicas adjuntado en el anexo "Preinscripciones Técnicas de los Puntos de Autoservicio".

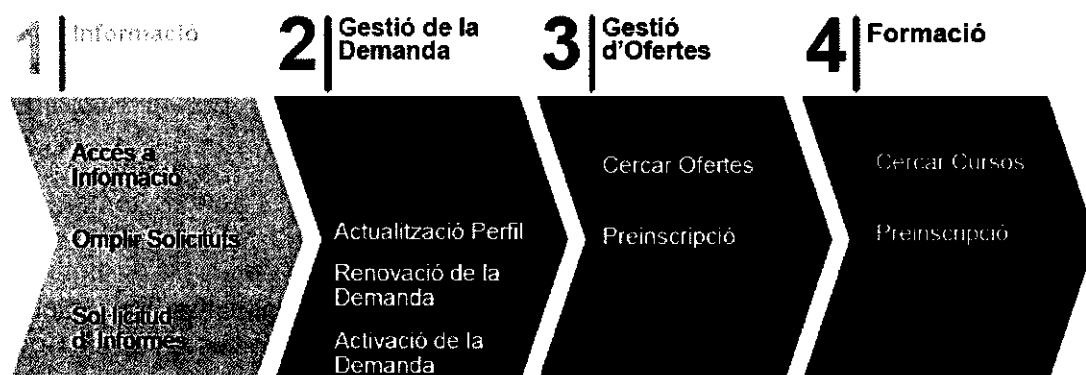


Figura 3: Funcionalidades de los Puntos de Autoservicio

Así pues, esta serie de funcionalidades que se observan en la imagen anterior permitirán al Ciudadano:

- Acceder a la información sobre los servicios ofrecidos por el SOC y por tanto conocer la información que debe entregar en cada uno de estos servicios y las prestaciones a las que puede acceder.
- Rellenar una serie de formularios y solicitudes tales como:
 - Solicitud simplificada de prestación de paro.
 - Programa de renta activa de inserción.
 - Prestación contributiva.
 - Subsidio por desempleo.
- Solicitar la impresión de ciertos informes tales como
 - DARDO (informe de renovación de la demanda).
 - Curriculum.
 - Servicios a los que está inscrito.
 - Informe de prestaciones.
 - Periodo de inscripción de demandante.

Por otra parte estos puntos de Autoservicio deben ofrecer una serie de funcionalidades que permitan a los ciudadanos gestionar la demanda de ofertas de empleo. Estas funcionalidades permitirán al ciudadano.

- Actualizar el perfil y los datos personales que el SOC dispone de su persona.
- Renovar su demanda de empleo en el SOC.
- Activar su demanda de empleo en el SOC en el caso que el estado de la misma sea "Bajo".
- Realizar una inscripción abreviada de demanda de empleo. Esta inscripción abreviada no es más que una preinscripción en el SOC, que les permite acceder a todos los servicios del sistema exceptuando aquellos que estén relacionados con los procesos de intermediación.

Estos puntos de Autoservicio deben disponer también de una serie de funcionalidades encaminadas a la gestión de ofertas de empleo. Así pues, desde estos puntos de Autoservicio, el ciudadano podrá:

- Buscar las ofertas de trabajo ofrece el SOC.
- Realizar la preinscripción en una de estas ofertas.

Del mismo modo, el sistema dispone de una serie de funcionalidades destinadas a gestionar los servicios de formación ofrecidos por el SOC. Estas funcionalidades permitirán al ciudadano:

- Buscar cursos de formación ofrecidos por el SOC.
- Realizar la preinscripción en uno de estos cursos.

Por otra parte, para cada una de estas funcionalidades, el punto de Autoservicio debe disponer de un sistema de ayuda para resolver las dudas que el usuario del mismo pueda tener.

3.2 Requisitos no Funcionales

3.2.1 Requisitos de Look & Feel

Las particularidades de apariencia que el Sistema debe cumplir son las siguientes:

- El logotipo de la empresa tiene que aparecer en todas las pantallas dando de este modo una sensación de seguridad, autoridad y confianza a los usuarios.
- Las pantallas usaran un estilo visual característico del SOC (acorde con su imagen) y que no sea molesto a la vista para el usuario, con la finalidad de que el sistema sea lo mas agradable posible y que los usuarios no sean reacios a utilizarlo.
- El diseño de las pantallas debe mostrar además un estilo autoritario y que fomente la confianza en el uso del sistema. Este estilo autoritario debe ayudar a solventar las dudas y reticencias que acostumbran a mostrar los ciudadanos de a pie con la automatización de procesos.

3.2.2 Requisitos de Usabilidad y Accesibilidad

Como se comenta en el documento de especificaciones técnicas elaborado con la colaboración del cliente, el cumplimiento de una serie de requisitos de usabilidad condicionará en gran medida el éxito del proyecto. Dado que los puntos de Autoservicio serán habitualmente utilizados por personas que no esta excesivamente familiarizados con este tipo de sistemas, se antoja indispensable que la interficie del sistema sea fácil de utilizar. Con el fin de alcanzar este objetivo, dicha interficie deberá cumplir una serie de requisitos que se detallan a continuación:

- El sistema utilizara una interfaz homogénea en todas las pantallas para facilitar el uso del mismo.
- La interficie estará formada por pantallas muy ligeras, es decir, con poca información simultánea.
- El sistema tendrá que poder usarse de forma intuitiva, dado que a ningún usuario le gusta leerse pesados manuales para poder utilizarlo. De éste modo, cualquier persona sin conocimientos previos de informática podrá utilizar el sistema sin necesitar una formación adicional.
- Se utilizarán imágenes e iconos de reconocimiento internacional en lugar de textos.
- Con respecto a formularios que el usuario podrá rellenar utilizando el punto de Autoservicio, el sistema no deberá pedir al usuario datos redundantes, conocidos o accesibles por otros medios.
- El sistema mostrará mensajes y textos descriptivos con la finalidad de que el usuario pueda comprender en todo momento la información que proporciona y requiere el sistema.

- El sistema permitirá el acceso a cualquier funcionalidad en cualquier momento. Adicionalmente destacará, en todo momento, el punto en el que se encuentra el usuario y permitirá que un usuario pueda volver a un punto anterior para ver o modificar datos anteriores.
- El sistema rellenará la mayoría de campos de datos mediante opciones, de este modo se acota el riesgo que un usuario cometa errores. En el caso que existan campos que no se rellenen de dicha forma, el sistema comprobará que no existan errores en la introducción de datos, informando, en caso de error, del tipo de error cometido.
- El sistema informará siempre del resultado de las operaciones realizadas. De éste modo el usuario tendrá certeza de que la operación realizada se ha producido del modo esperado.
- El sistema se podrá utilizar en castellano y catalán indistintamente. El usuario se encargará de seleccionar el idioma que desee, de éste modo, se facilitará en gran medida la comprensión del usuario.
- El sistema deberá esconder al usuario cualquier detalle asociado a su implementación, es decir, el usuario deberá interactuar con el sistema como si de una caja negra se tratase.
- El identificador de usuario debe aparecer en todas las pantallas siempre que el usuario haya iniciado una sesión en el sistema.
- El sistema debe poder ser utilizado por usuarios con deficiencias visuales.

3.2.3 Requisitos de Rendimiento, Precisión, Escalabilidad y Mantenibilidad

Por lo que respecta al rendimiento del sistema, su mantenibilidad y escalabilidad, se deben cumplir los siguientes requisitos:

- El tiempo de respuesta del sistema para cualquier petición del usuario no será lo suficientemente elevado como para provocar que el usuario piense que se ha producido un error y que el sistema no responde.
- Todas las cantidades monetarias que aparezcan por pantalla tendrá una precisión de dos decimales.
- El formato utilizado para mostrar las fechas debe ser el mismo en todo el sistema (dd/MM/yyyy)
- El sistema deberá ser capaz de operar durante los 5 minutos posteriores a su desconexión de la red eléctrica.
- Dado que está prevista la instalación de 219 Puntos de Autoservicio en las distintas oficinas del SOC, el sistema deberá soportar un mínimo de 500 conexiones simultáneas por segundo. Este número viene determinado por la posibilidad de que en el futuro se amplíe el número de cajeros y por un futuro acceso vía Web al sistema.

- Se debe desarrollar un sistema modular que permita el crecimiento tanto en aspectos de maquinaria (nuevos puntos de Autoservicio) como en volumen de datos de forma sencilla, rápida y económica.
- El Sistema debe tener la capacidad para estar operativo las 24 horas, 7 días a la semana. Este requisito de nuevo viene determinado por un futuro acceso vía Web al sistema.
- El Sistema debe ser capaz de interaccionar con los datos asociados al sistema SICAS, sistema ya existente y con el que actualmente trabaja el SOC.

3.2.4 Requisitos de Seguridad y Fiabilidad

Teniendo en cuenta lo que respecta a la seguridad y fiabilidad del Sistema, se deben tener en consideración los siguientes criterios:

- Para poder acceder a ciertas funcionalidades del Sistema, los usuarios deberán estar registrados y autenticados en el mismo.
- La información privada de cada usuario sólo podrá ser modificada por el propio usuario ya que en todo momento se debe mantener la privacidad de los usuarios registrados en el Sistema.
- El Sistema deberá estar protegido contra el software malintencionado así como de usuarios malintencionados. De éste modo se evita la pérdida, fuga, corrupción o alteración de los datos almacenados y se garantiza la privacidad de los mismos.
- El sistema deberá cumplir con las normas de seguridad de la información ISO/IEC/UNE 17799 y con UNE1502 de gestión de la seguridad de la información, adaptadas a la estructura administrativa, personal y entorno tecnológico del cliente y aplicadas de forma proporcional a los riesgos reales.
- Todas las transacciones que se efectúen en el Sistema estarán encriptadas de forma segura. Esto es debido a la transmisión de datos personales que no deben ser accesibles para usuarios malintencionados.

3.2.5 Requisitos Legales

Por último hay que tener en cuenta una serie de requisitos legales de vital importancia para nuestro sistema, debido a los datos personales que almacena. Por ello, el sistema deberá cumplir el Reglamento de Seguridad del Real Decreto 994/1999 de la Ley Orgánica de Protección de Datos Personales (LOPD).

3.3 Mejoras Añadidas

En este apartado se describen una serie de mejoras o ampliaciones propuestas a las funcionalidades requeridas por el SOC en el documento de preinscripciones técnicas. Estas mejoras están relacionadas con la gestión de las cuentas de los ciudadanos que interactúan con el sistema. Se propone:

- Definir una funcionalidad que permita a los Ciudadanos crear una cuenta en el sistema que les dé acceso a todos los servicios y funcionalidades del mismo, incluyendo los procesos de intermediación, procesos que no eran accesibles mediante la creación de una cuenta utilizando la funcionalidad de inscripción abreviada.
- Definir una funcionalidad que permita a los Ciudadanos eliminar su cuenta del sistema, tanto si esta cuenta se ha creado utilizando la funcionalidad asociada a la inscripción abreviada como la funcionalidad asociada a la mejora anteriormente propuesta.

3.4 Anexos asociados

En el anexo “**Preinscripciones Técnicas de los Puntos de Autoservicio**” adjuntado en el final de la memoria se puede consultar la definición de las características realizada por el SOC que deben cumplir los Puntos de Autoservicio.

Por otra parte, el anexo “**Supplementary Specifications**” contiene una descripción más detallada de cada uno de los requisitos no funcionales del Sistema.

4 Business Model

4.1 Introducción

El objetivo fundamental que cumple un sistema de información en una empresa es automatizar una serie de procesos de negocio. Entendemos por proceso de negocio el conjunto de actividades realizadas por trabajadores de la empresa, los eventos que provocan la realización de estas actividades, así como los elementos o entidades que deben gestionar y crear estos trabajadores con el objetivo de producir un resultado destinado a satisfacer las necesidades de un cliente o entidad externa a la empresa.

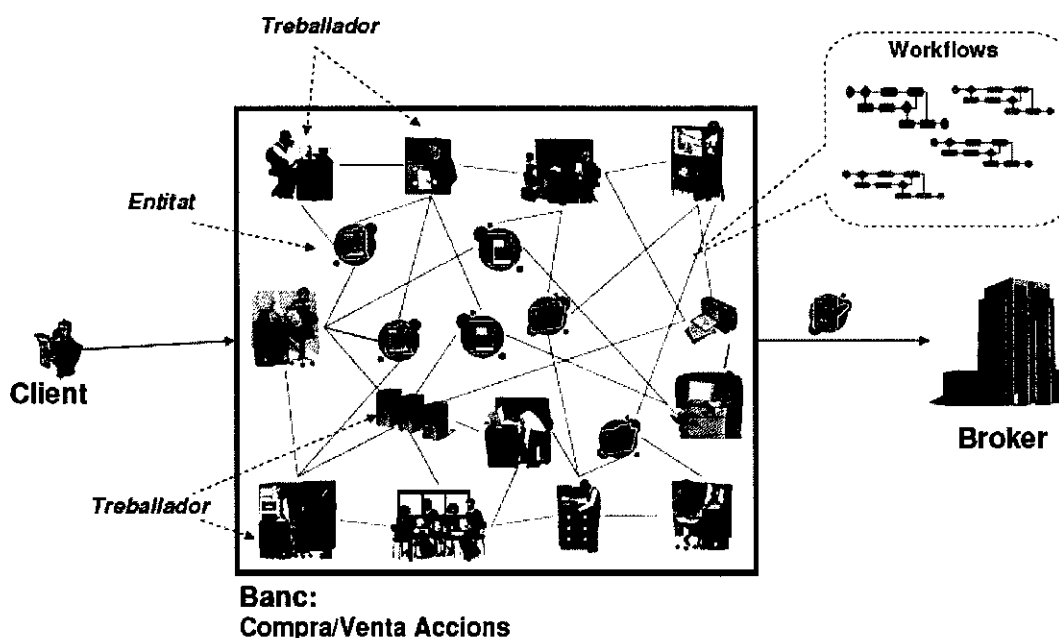


Figura 4: Modelado de Negocio

El objetivo de modelar los procesos de negocio del *Servei d'Ocupació de Catalunya* para la realización del proyecto que nos atañe es conocer y comprender cada una de las actividades que lleva a cabo dicha empresa con el fin de determinar de una manera más precisa y eficaz las necesidades que los Puntos o Terminales de Autoservicio deben satisfacer.

Así pues, este conocimiento profundo de dichos procesos de negocio debe facilitar la realización de tareas posteriores asociadas al análisis y diseño de las funcionalidades asociadas a los Puntos de Autoservicio, así como a aportar una visión común y unificada de estos procesos de negocio a cada uno de los integrantes del proyecto.

Para realizar este modelado de los procesos de negocio se analizará el *Servei d'Ocupació de Catalunya* como un sistema en el que se definirán sus aspectos estáticos y dinámicos, tanto internos como externos, utilizando diagramas de UML.

Para realizar este modelado de los procesos de negocio se ha procedido a identificar las actividades asociadas a estos procesos, los trabajadores o *workers* del SOC que intervienen en la realización de dichas actividades, así como las entidades de negocio que se obtienen y/o manipulan durante la realización de estos procesos de negocio. Para ello se definirán:

- **Business Actors**
- **Business Use Cases**
- **Business Object Model**
- **Business Use Case Realization**

4.2 Business Actors

El modelado de los **Business Actors** corresponde a la definición de una serie de roles interpretados por personas o entidades ajenas al Sistema y que interactúan con él.

La definición de estos **Business Actors** facilitará en gran medida ciertas tareas futuras asociadas al proceso de análisis tales como la definición de los actores de los casos de uso del Sistema de Puntos de Autoservicio, es decir, la definición de las diferentes tipologías de usuarios que interaccionarán con dichos Puntos de Autoservicio.

Considerando los diferentes procesos de negocio asociados al *Servei d'Ocupació de Catalunya*, se han definido los siguientes **Business Actors**:

- **Ciudadano:** rol que corresponde a los ciudadanos de a pie a los cuales van destinados cada uno de los servicios ofrecidos por el SOC. Se puede afirmar, por tanto, que los ciudadanos de a pie son de alguna manera los clientes del SOC.
- **Ciudadano Inscrito:** rol que corresponde a los ciudadanos de a pie a los cuales se destinan los servicios ofrecidos por el SOC, pero con la particularidad de que este rol está asociado a ciudadanos que están inscritos en el SOC mediante el procedimiento de *Inscripción Abreviada*.
- **Ciudadano Inscrito:** rol que corresponde a los ciudadanos de a pie a los cuales se destinan los servicios ofrecidos por el SOC, pero con la particularidad de que este rol está asociado a ciudadanos que están inscritos en el SOC mediante el procedimiento de *Inscripción Abreviada*. Esta inscripción les permite acceder a funcionalidades que están vetadas para ciudadanos no inscritos.
- **Ciudadano Registrado:** se trata de un rol muy similar al anterior, pero con la particularidad de que los ciudadanos no se han inscrito en el Sistema mediante el proceso de *Inscripción Abreviada*, sino a través de la creación de una cuenta. Esta variación permite que los ciudadanos asociados a este rol pueden acceder a funcionalidades del SOC asociadas a servicios de intermediación laboral.
- **Ofertante Empleo:** se trata de un rol que representa a aquellas empresas que ofrecen ofertas de empleo a los Ciudadanos a través del SOC. Se puede

afirmar que, de alguna manera, este rol representa a uno de los principales proveedores del SOC, ya que le proporciona las ofertas de empleo que después serán ofrecidas a los ciudadanos.

- **Ofertante Curso de Formación:** rol que representa a aquellas entidades que ofrecen a los ciudadanos Cursos de Formación a través del SOC. Al igual que con el rol **Ofertante Empleo**, se puede afirmar que el rol **Ofertante Curso de Formación** representa a otro de los principales proveedores del SOC, debido a que le suministra los Cursos de Formación que más tarde serán ofrecidos a los ciudadanos.

En la siguiente imagen, se puede observar las diferentes relaciones que se establecen entre los **Business Actors** definidos.

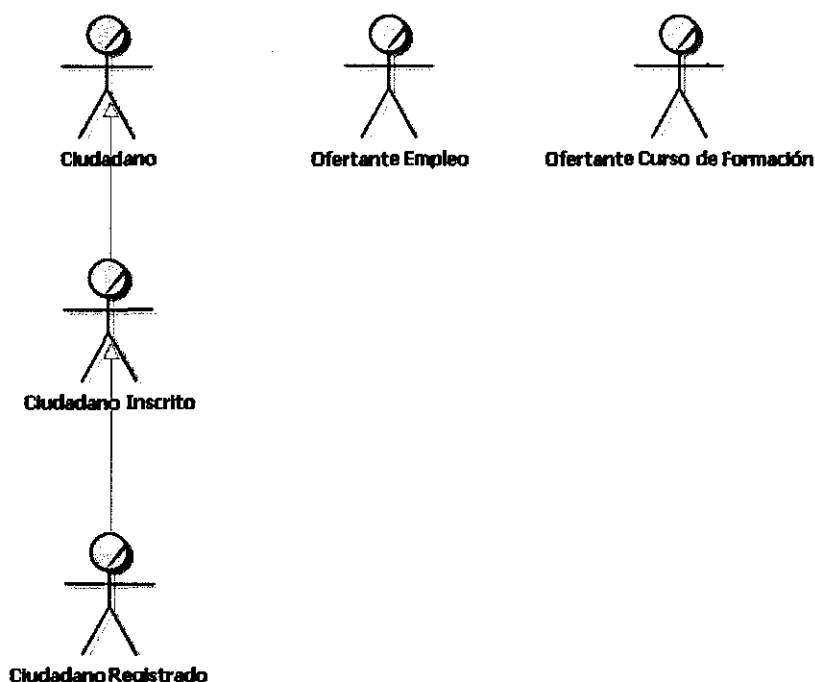


Figura 5. Business Actors

Como se puede observar en esta imagen, se establece una relación de herencia entre los diferentes roles asociados a los ciudadanos que acceden y utilizan los servicios ofrecidos por el SOC. Esta relación de herencia viene determinada por el hecho de que ciertos servicios proporcionados por el SOC están reservados a ciudadanos inscritos o registrados en el *Servei d'Ocupació de Catalunya*.

4.3 Business Use Cases

Una vez definidos los diferentes actores que pueden interaccionar con los servicios ofrecidos por el SOC, es el momento de definir concretamente que servicios ofrece este negocio, es decir, definir una agrupación de diferentes procesos de negocio de una determinada tipología vistos desde un punto de vista externo.

Esta modelización de los *Business Use Cases* facilitará en un futuro cercano la definición de los diferentes Casos de Uso que definen el Sistema de Puntos de Autoservicio.

Para definir, crear y diferenciar los diferentes casos de uso que conforman el *Business Model* se ha seguido un criterio lógico de utilización. Se han clasificado las distintas actividades de negocio en función de su aportación a la empresa, los actores que participan en estas actividades y el fin u objetivo de las mismas.

En primer lugar, cabe destacar una serie de procesos asociadas a la gestión y mantenimiento de los datos de los ciudadanos que interaccionan con el Sistema accediendo a los distintos servicios ofrecidos por el SOC. Esta serie de actividades o procesos han sido agrupados en el *Business Use Case Gestión Cuenta*.

Por otra parte, cabe remarcar también la presencia de un conjunto de procesos destinados a informar a los ciudadanos sobre las actividades y servicios ofrecidos por el SOC, así como al intercambio de información entre el SOC y los propios ciudadanos. Estas actividades han sido agrupadas en el *Business Use Case Gestión Información*.

Como ya se ha comentado previamente a lo largo de esta documentación, existen una serie de procesos de negocio encaminados a proporcionar un enriquecimiento de los conocimientos de los ciudadanos mediante una amplia oferta de Cursos de Formación. Actividades como la búsqueda de Cursos, la publicación de nuevos Cursos o la inscripción de un ciudadano en uno de estos Cursos conforman el *Business Use Case Gestión Cursos*.

Lógicamente, dada la naturaleza del *Servei d'Ocupació de Catalunya*, encontramos también una serie de procesos de negocio que podrían calificarse como fundamentales y que están relacionados con las Ofertas de Empleo destinadas a los ciudadanos de a pie. Así pues, actividades como la búsqueda, inscripción o publicación de Ofertas de Empleo conforman el *Business Use Case Gestión Ofertas*.

Finalmente, cabe destacar también la presencia de una serie de procesos de negocio relacionados con el soporte y atención al ciudadano. Se trata de una serie de actividades destinadas a solucionar y ayudar al ciudadano ante cualquier eventualidad o problema derivado de su interacción con los servicios y funcionalidades ofrecidos por el SOC. Esta serie de procesos de negocio determinan el *Business Use Case Soporte y Atención al Cliente*.

En las siguientes imágenes se pueden observar las diferentes asociaciones que se establecen entre los diferentes Casos de Uso del Modelo de Negocio y los *Business Actors*. A la hora de observar y evaluar estas asociaciones, cabe recordar las diferentes relaciones que se establecen entre los *Business Actors*.

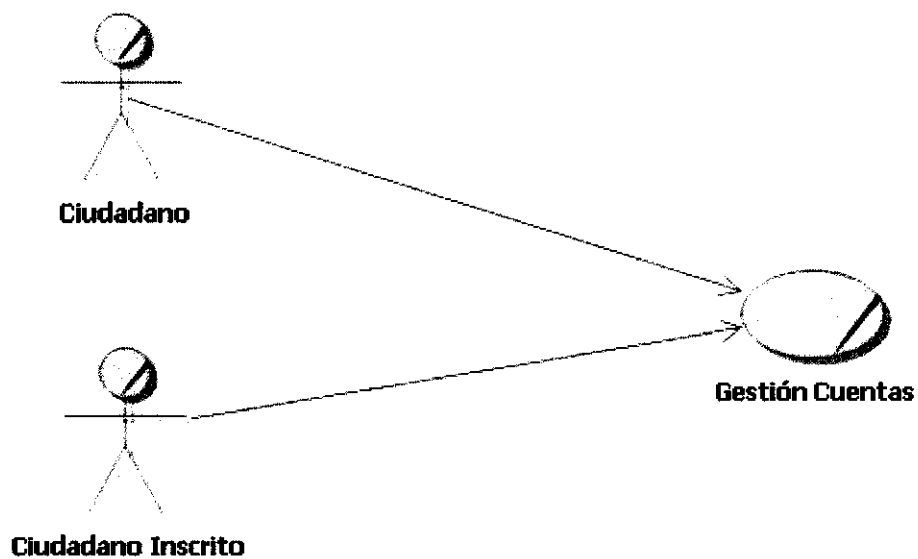


Figura 6. Business Use Case Gestión Cuentas

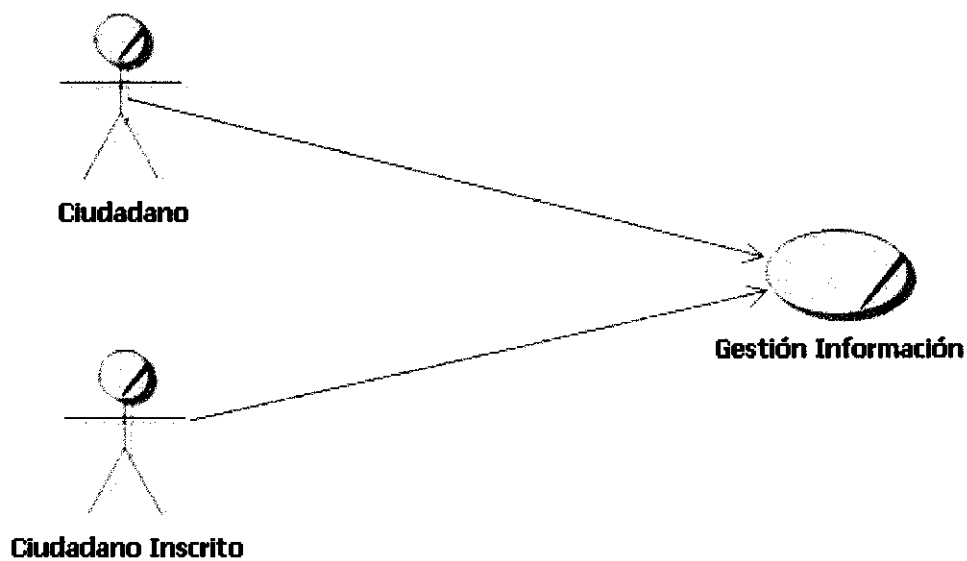


Figura 7. Business Use Case Gestión Información

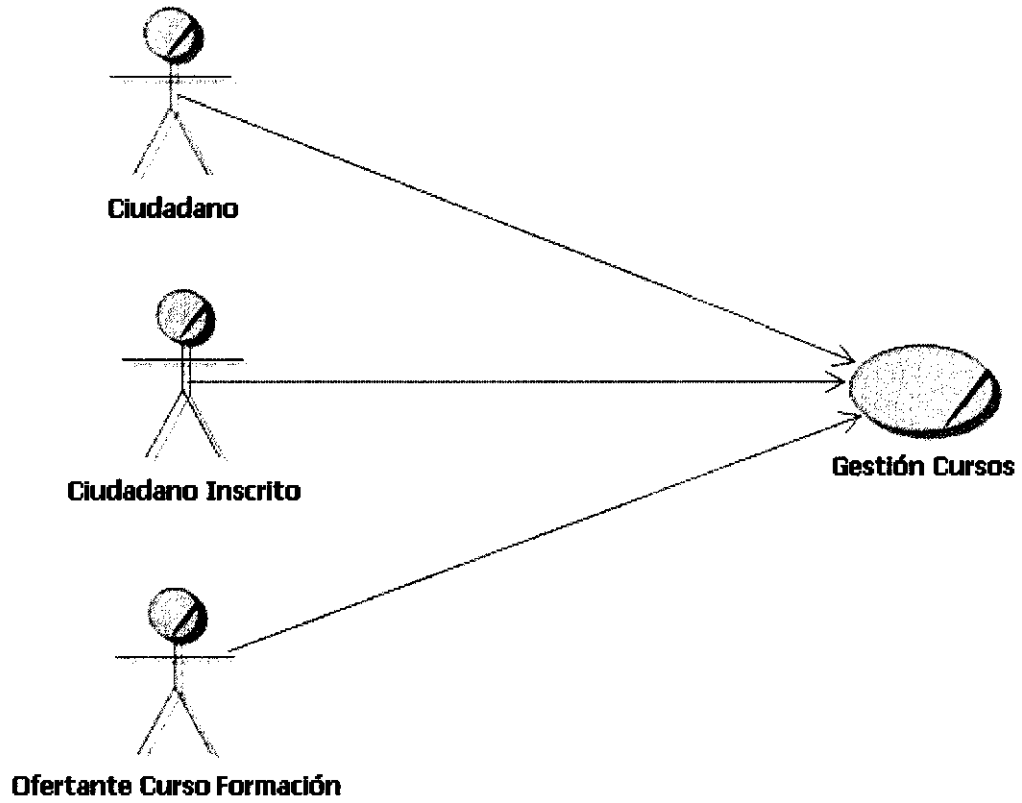


Figura 8. Business Use Case Gestión Cursos

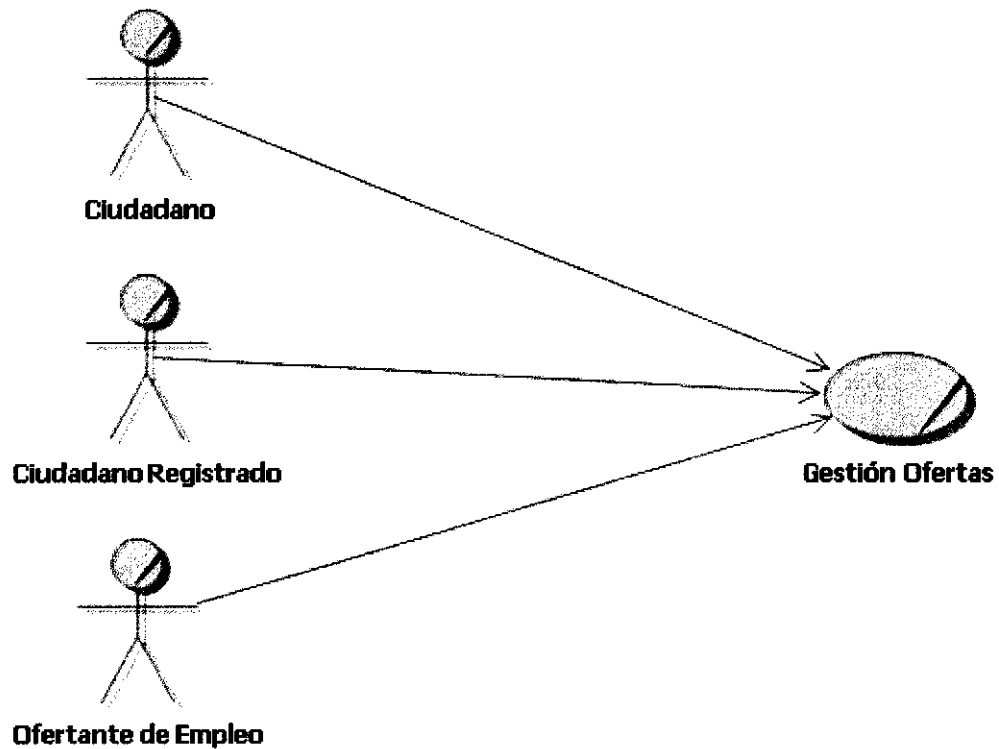


Figura 9. Business Use Case Gestión Ofertas

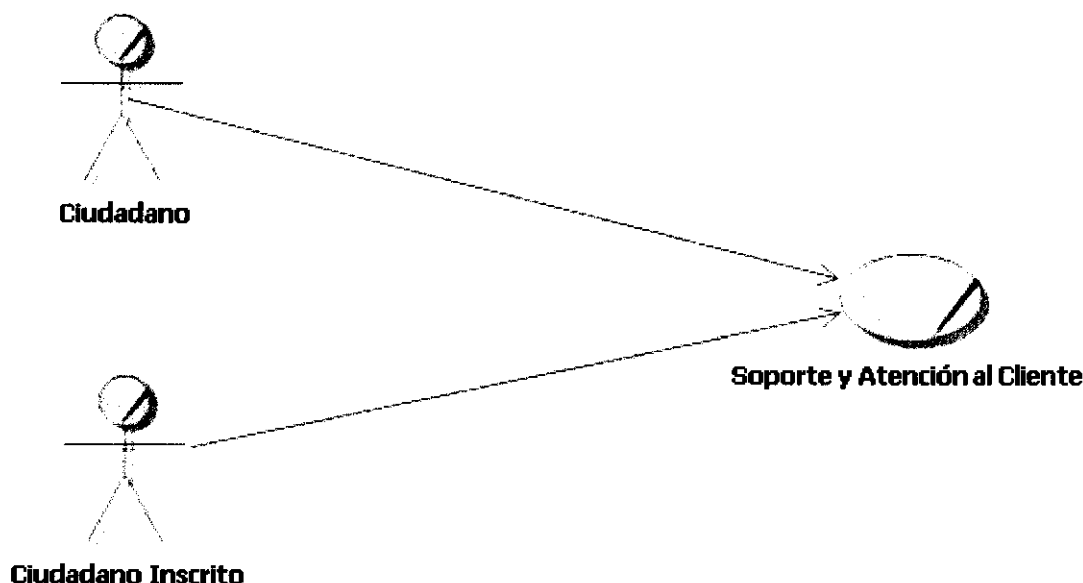


Figura 10. Business Use Case Soporte

4.4 Business Object Model

El **Business Object Model** corresponde a la definición de los trabajadores (**Workers**) y las entidades (**Entities**) que intervienen en el proceso de negocio que se pretende modelizar.

Se entiende por **Worker** la abstracción que representa el rol realizado por una/s persona/s o entidad/es que participa activamente en el desarrollo del proceso de negocio.

Una **Entity** en cambio representa la información que intercambian los **Workers** entre si y con los **Business Actors** durante el desarrollo del proceso de negocio.

Para la definición de los **Business Objects** en el modelado del proceso de negocio del *Servei d'Ocupació de Catalunya* se ha procedido a realizar una división departamental del SOC, división departamental que en el ámbito del **Business Model** es conocida como **división en Business Systems**. Esta división se ha basado principalmente en criterios asociados a la temática de los procesos de negocio, agrupando por tanto procesos de una temática similar en un mismo departamento o **Business System**. En base a estos criterios se han definido los siguientes **Business Systems**:

- **Administración:** *Business System* encargado de toda la información asociada a los Ciudadanos Inscritos o Registrados en el SOC.
- **Formación:** *Business System* encargado de gestionar toda la información asociada a los Cursos de Formación ofrecidos por el SOC.
- **Intermediación:** *Business System* encargado de realizar todos los trámites asociados a las Ofertas de Empleo ofrecidas por el SOC.

- **Información:** *Business System* encargado de atender todas las peticiones de los usuarios del SOC relacionadas con el acceso de éstos a los distintos procesos informativos que realiza el SOC.
- **Soporte:** *Business System* encargado de atender las peticiones de ayuda y quejas realizadas por los ciudadanos.

4.4.1 *Entities y Workers del Business System de Administración*

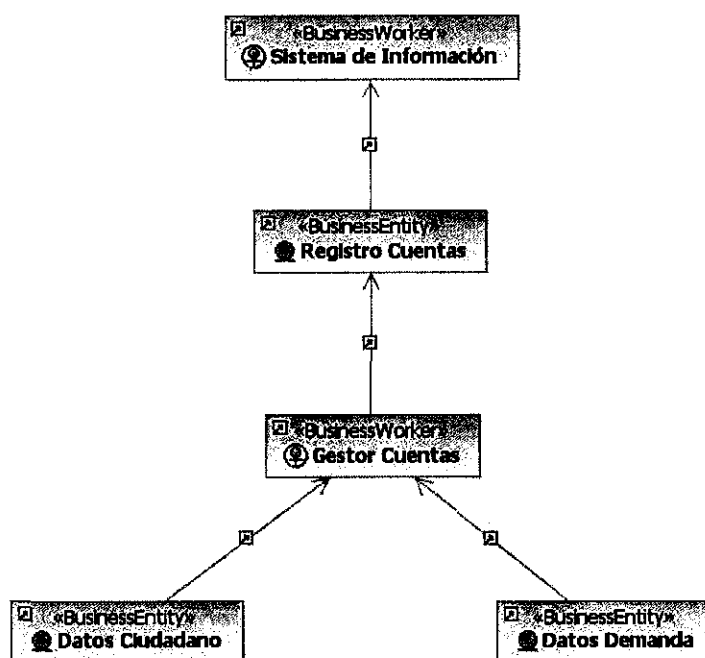


Figura 11. Entities y workers Administración

El *worker* Gestor Cuentas recibe y valida los Datos de los Ciudadanos y sus demandas de empleo. A partir de estos datos, y tras realizar la validación pertinente, genera los Registros de Cuentas que contienen toda la información personal de un Ciudadano que a continuación se registra en un sistema ya existente.

4.4.2 *Entities y Workers del Business System de Formación*

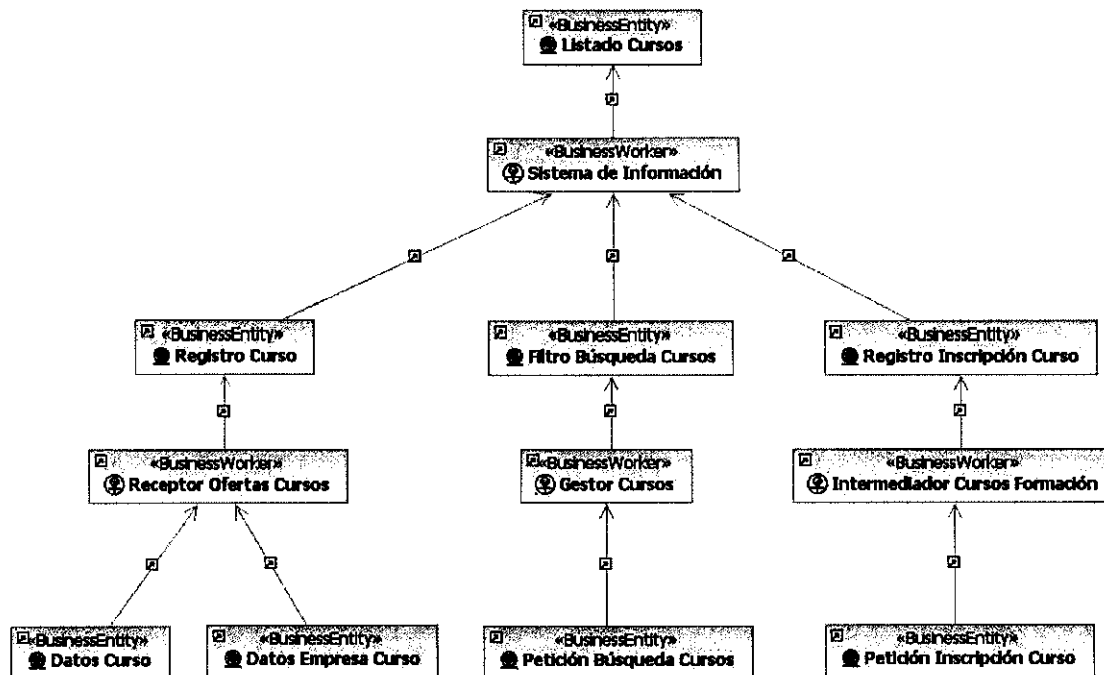


Figura 12. Entities y workers Formación

En primer lugar cabe destacar la figura del *worker* Receptor Ofertas Cursos que se encarga de recibir y registrar toda la información asociada a la publicación de un nuevo Curso de Formación. El Receptor Ofertas Cursos recibirá y validará los Datos del Curso, así como la información de la empresa que impartirá dicho Curso de Formación. Tras validar esta información, generará el Registro de un nuevo Curso de Formación que se inserta en el sistema ya existente para ponerlo a disposición de los ciudadanos.

Por otra parte, cabe remarcar que este *Business System* se encarga también de atender las peticiones de búsqueda de Cursos por parte de los ciudadanos. Un ciudadano realizará una Petición de Búsqueda de Cursos que será atendida por el *worker* Gestor Cursos. A partir de esta petición, en *worker* generará un filtro de búsqueda que introducirá en el Sistema de Información ya existente y éste le retornará un Listado de Cursos que cumplen las características especificadas en la petición.

Finalmente, este *Business System* de Formación se encarga también de realizar los trámites asociados a la inscripción de un ciudadano en Curso de Formación. El ciudadano solicitará inscribirse en un Curso de Formación. Un *worker* llamado Intermediador Cursos Formación atenderá esta petición y generará una inscripción para el Curso en cuestión que será registrada en el Sistema de Información ya existente.

4.4.3 Entities y Workers del Business System de Intermediación

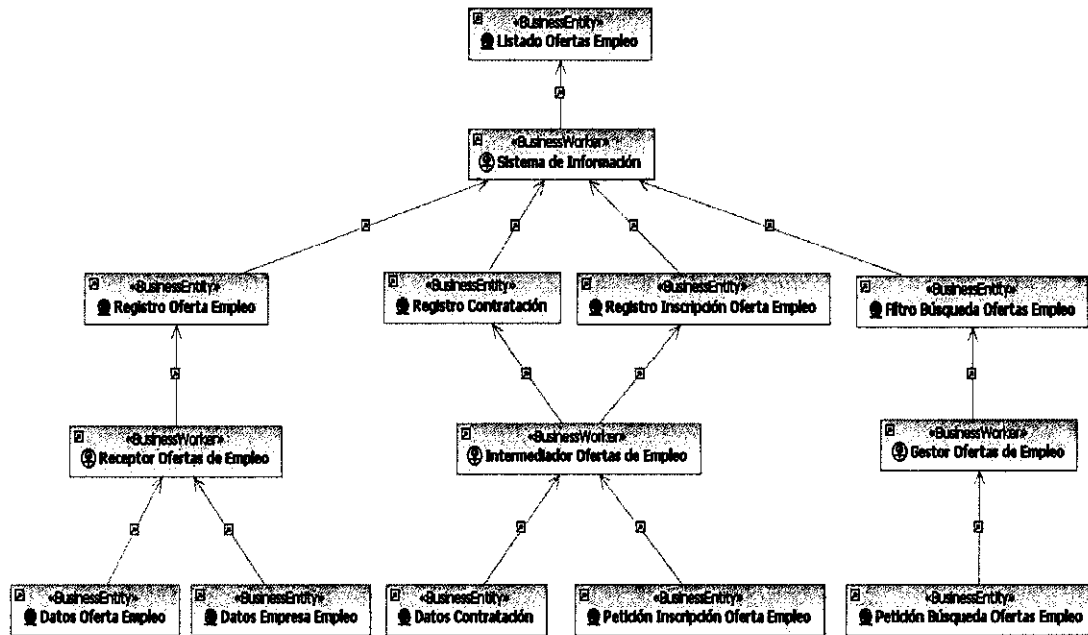


Figura 13. Entities y workers de Intermediación

En primer lugar, cabe destacar al *worker* Receptor Ofertas de Empleo. Este *worker* será el encargado de atender las nuevas publicaciones de Ofertas de Empleo por parte de las empresas que se ponen en contacto con el SOC. A partir de los Datos de la Oferta de Empleo y los Datos Empresa de Empleo, y tras realizar la validación pertinente, este *worker* generará un nuevo Registro de una Oferta de Empleo que se registrará en el Sistema de Información ya existente.

Por otra parte, este *Business System* y más concretamente el *worker* Intermediador Ofertas de Empleo se encargará de atender las Peticiones de Inscripción en Ofertas por parte de los ciudadanos. A partir de esta petición, el *worker* en cuestión genera un Registro de Inscripción que introduce en el Sistema de Información. Además, este mismo *worker* será el encargado de registrar los procesos de intermediación realizados con éxito, es decir, cuando tras una inscripción en una Oferta de Empleo realizada a través del SOC se produce una contratación del ciudadano que realizó la inscripción por parte de la empresa asociada a dicha Oferta de Empleo, se notifica al SOC dicha contratación y este *worker* registrará la contratación en el Sistema de Información.

Finalmente, destacar que este *Business System* de Intermediación se encarga de atender las peticiones de Búsqueda de Ofertas de Empleo por parte de los Ciudadanos. El *worker* Gestor Ofertas de Empleo atenderá estas peticiones y generará un Filtro de Búsqueda que introducirá en el Sistema de Información ya existente, el cual le retornará un Listado de Ofertas de Empleo.

4.4.4 Entities y Workers del Business System de Información

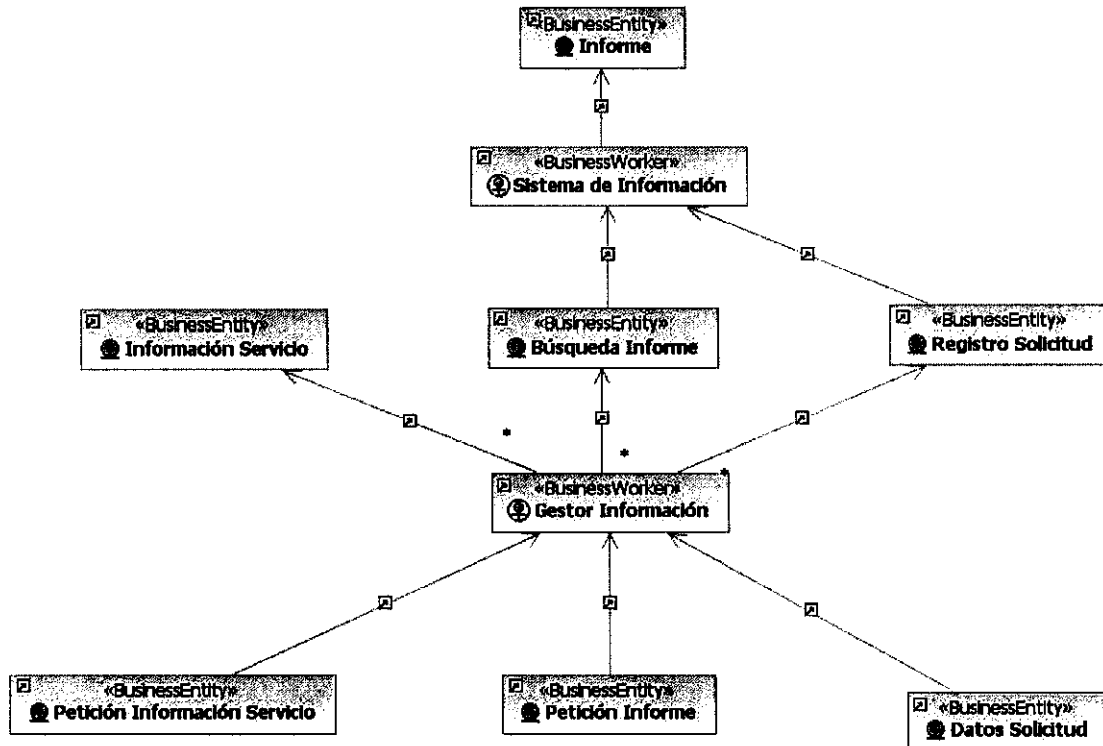


Figura 14. Entites y workers de Información

En primer lugar, este departamento permite que los ciudadanos puedan informarse de los distintos servicios ofrecidos por el SOC. Así pues, un ciudadano cualquiera podrá realizar una Petición de Información Servicio que será atendida por el *worker* Gestor Información, el cual le proporcionará directamente una serie de documentos informativos sobre dicho servicio.

Por otra parte, los ciudadanos inscritos o registrados en el SOC podrán acceder a este departamento para solicitar la entrega de una serie de informes asociados a los datos que el SOC dispone de ellos. Así pues, un ciudadano inscrito o registrado podrá realizar una Petición de un Informe. Esta petición será atendida por el *worker* Gestor Información que realizará una búsqueda en el Sistema de Información ya existente de dicho informe y finalmente el Sistema proporcionará el informe buscado.

Finalmente, este *Business System* permitirá a cualquier ciudadano rellenar una solicitud para acceder a una serie de servicios ofrecidos por el SOC. Un ciudadano proporcionar una serie de datos asociados a esa solicitud, el *worker* Gestor Información validará esos datos y generará el Registro de Solicitud que se almacenará en el Sistema de Información ya existente.

4.4.5 Entities y Workers del Business System de Soporte

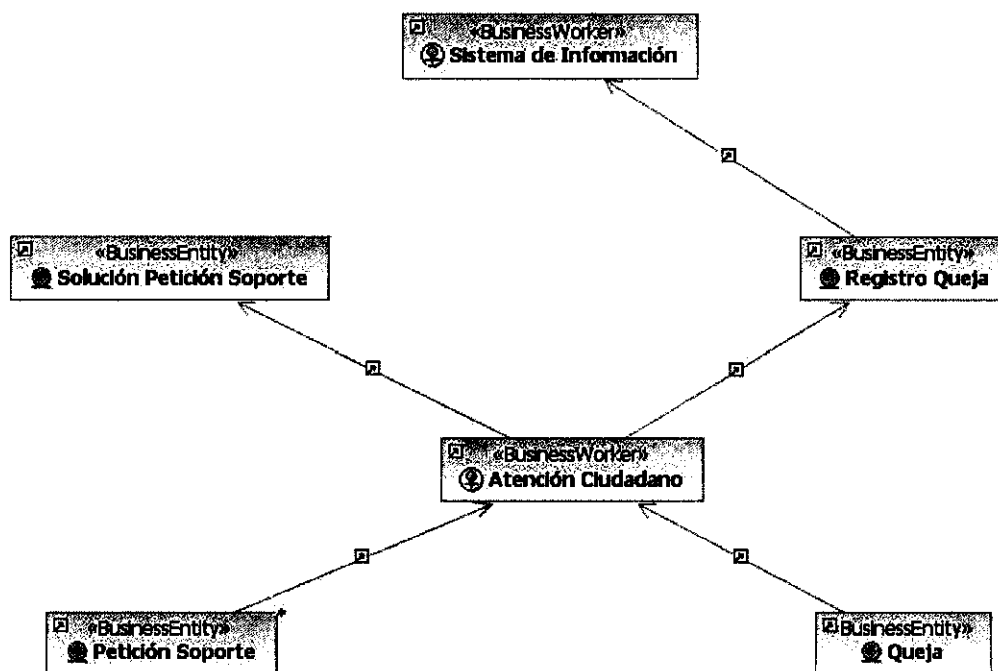


Figura 15. Entites y workers de Soporte

El *worker* Atención Ciudadano será el encargado de atender las Peticiones de Soporte de los ciudadanos y proporcionarles una solución al problema. Además, será el encargado de atender las posibles quejas que puedan tener los Ciudadanos sobre el SOC o las empresas que en él ofertan Cursos de Formación y Ofertas de Empleo. Tras validar esta queja, generará un Registro que introducirá en el Sistema de Información ya existente.

4.5 Business Use Case Realitization

La modelización de las *Business Use Case Realitization* pretende realizar una descripción de cómo colaboran internamente los *Business Workers* y las *Business Entities* para llevar a cabo cada uno de los procesos de negocio que forman parte de los *Business Use Cases* definidos, es decir, cada *Business Use Case Realitization* pretende realizar una modelización de cada uno de los procesos de negocio del SOC. Esta modelización de los procesos de negocio se realizará mediante diagramas de actividad UML.

Esta modelización de los procesos de negocio mediante diagramas UML debe aportar una visión detallada de cada una de las actividades que se llevan a cabo en el *Servei d'Ocupació de Catalunya*, facilitando de este modo el análisis y evaluación de las necesidades que el Sistema de Puntos de Autoservicio debe satisfacer.

Así pues, para cada uno de los *Business Use Cases* especificados anteriormente, se han definido una serie de *Business Use Case Realizations*.

Para el *Business Use Case Gestión Cuenta*, se han definido las siguientes realizaciones o procesos de negocio:

- **Inscripción Completa:** proceso de negocio asociado a la inscripción de un ciudadano en el SOC dándole acceso a todas las funcionalidades y servicios ofrecidos incluyendo los servicios de intermediación laboral.
- **Eliminar Cuenta:** proceso de negocio que permite a los ciudadanos eliminar su inscripción en el SOC.
- **Modificar Datos:** proceso de negocio que permite que los ciudadanos modifiquen sus datos personales registrados en el SOC.
- **Renovación Demanda:** proceso de negocio que permite a los ciudadanos realizar una renovación periódica de su estado de demanda.
- **Activación Demanda:** proceso de negocio que permite a los ciudadanos realizar una activación de su demanda.
- **Inscripción Abreviada:** proceso de negocio asociado a la inscripción de un ciudadano en el SOC dándole acceso a todos los servicios ofrecidos por el SOC exceptuando los procesos de intermediación laboral.
- **Completar Demanda:** proceso de negocio que permite que ciudadanos que han realizado una Inscripción Abreviada, completen el proceso de inscripción y puedan acceder a procesos de intermediación laboral.

Para el *Business Use Case Gestión Información*, se han definido las siguientes realizaciones o procesos de negocio:

- **Consultar Información Servicio:** proceso de negocio destinado a informar a los ciudadanos sobre los servicios ofrecidos por el SOC.
- **Rellenar Solicitud:** proceso de negocio que permite a los ciudadanos rellenar una serie de solicitudes con el objetivo de acceder y beneficiarse de ciertos servicios ofrecidos por el SOC.
- **Petición de Informes:** proceso de negocio que permite que a los ciudadanos consultar e imprimir la información de una serie de informes que contienen sus datos personales y/o laborales.

Para el *Business Use Case Gestión Cursos*, se han definido las siguientes realizaciones o procesos de negocio:

- **Buscar Curso:** proceso de negocio que permite a los ciudadanos consultar la información de una serie de Cursos de Formación ofrecidos por el SOC.
- **Inscripción Curso:** proceso de negocio que permite a los ciudadanos inscribirse en los Cursos de Formación ofrecidos por el SOC.
- **Publicar Nuevo Curso:** proceso de negocio protagonizado por el *Business Actor Ofertante Curso de Formación* mediante el cual, dicho Ofertante ofrece al SOC la publicación de un nuevo Curso de Formación destinado a los ciudadanos.

Para el *Business Use Case Gestión Ofertas*, se han definido las siguientes realizaciones o procesos de negocio:

- **Buscar Oferta:** proceso de negocio que permite a los ciudadanos consultar la información de una serie de Ofertas de Empleo ofrecidas por el SOC.
- **Inscripción Oferta:** proceso de negocio que permite a los ciudadanos inscribirse en las Ofertas de Empleo ofrecidas por el SOC.

- **Publicar Nueva Oferta:** proceso de negocio protagonizado por el *Business Actor Ofertante de Empleo* mediante el cual, dicho Ofertante ofrece al SOC la publicación de una nueva Oferta de Empleo destinada a los ciudadanos.
- **Candidato Aceptado:** proceso de negocio mediante el cual un **Ofertante de Empleo** notifica al SOC que, uno de los ciudadanos que se han inscrito a través del SOC en una Oferta de Empleo publicada por dicho Ofertante, ha sido contratado para el puesto de trabajo ofrecido.

Para el *Business Use Case Soporte y Atención al Cliente*, se han definido las siguientes realizaciones o procesos de negocio:

- **Queja Oferta Trabajo/Curso Formación:** proceso de negocio que permite a los ciudadanos formular al SOC una queja sobre el funcionamiento o trato recibido en un Curso de Formación u Oferta de Empleo.
- **Soporte:** proceso de negocio mediante el cual los ciudadanos consultan al personal del SOC una serie de dudas asociadas a los servicios y funcionalidades ofrecidas con el objetivo de poder resolver dichas dudas.

Como ya se ha comentado previamente, para cada una de estas realizaciones se ha definido un diagrama de actividad UML. En las siguientes imágenes se pueden observar los diagramas correspondientes a las realizaciones **Inscripción Oferta** y **Rellenar Solicitud**, asociando a cada diagrama una breve descripción de las diferentes interacciones que se producen entre los actores participantes.

Inscripción Oferta

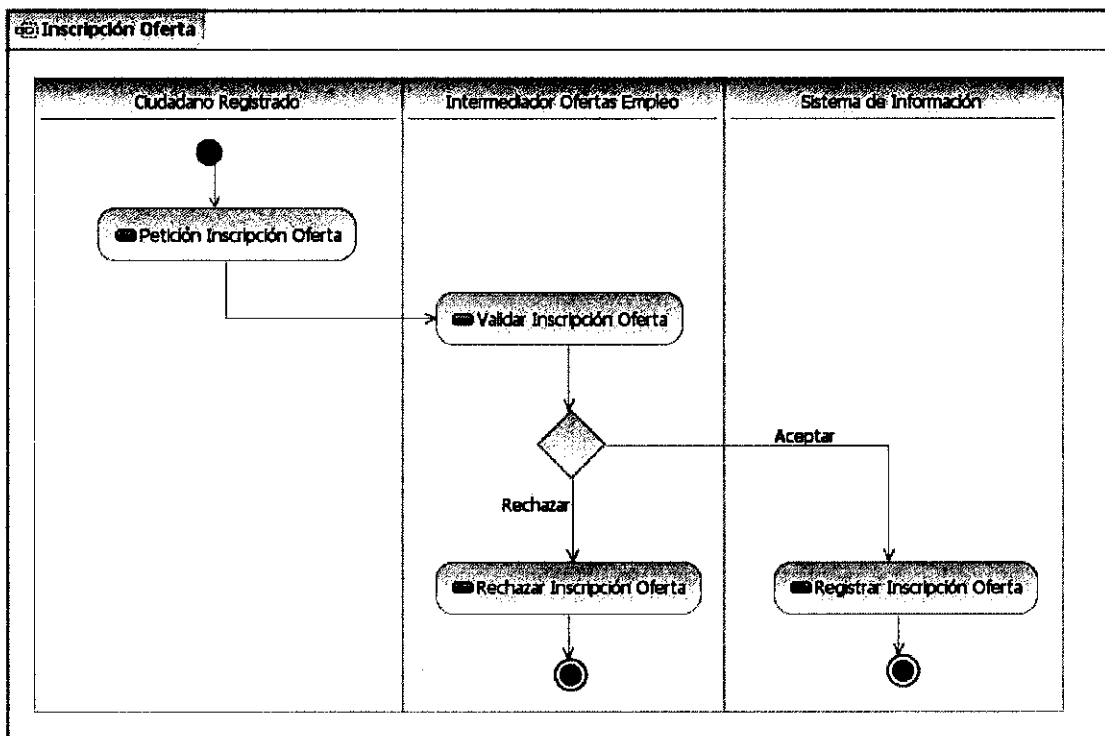


Figura 16. Business Use Case Realization Inscripción Oferta

1. Un Ciudadano Registrado solicita al SOC inscribirse en una Oferta de Empleo.
2. El Intermediador de Ofertas de Empleo valida la petición y en caso de que ésta sea correcta, procede a registrar dicha inscripción.
3. El Sistema de Información registra la inscripción del Ciudadano Registrado en la Oferta.

Rellenar Solicitud

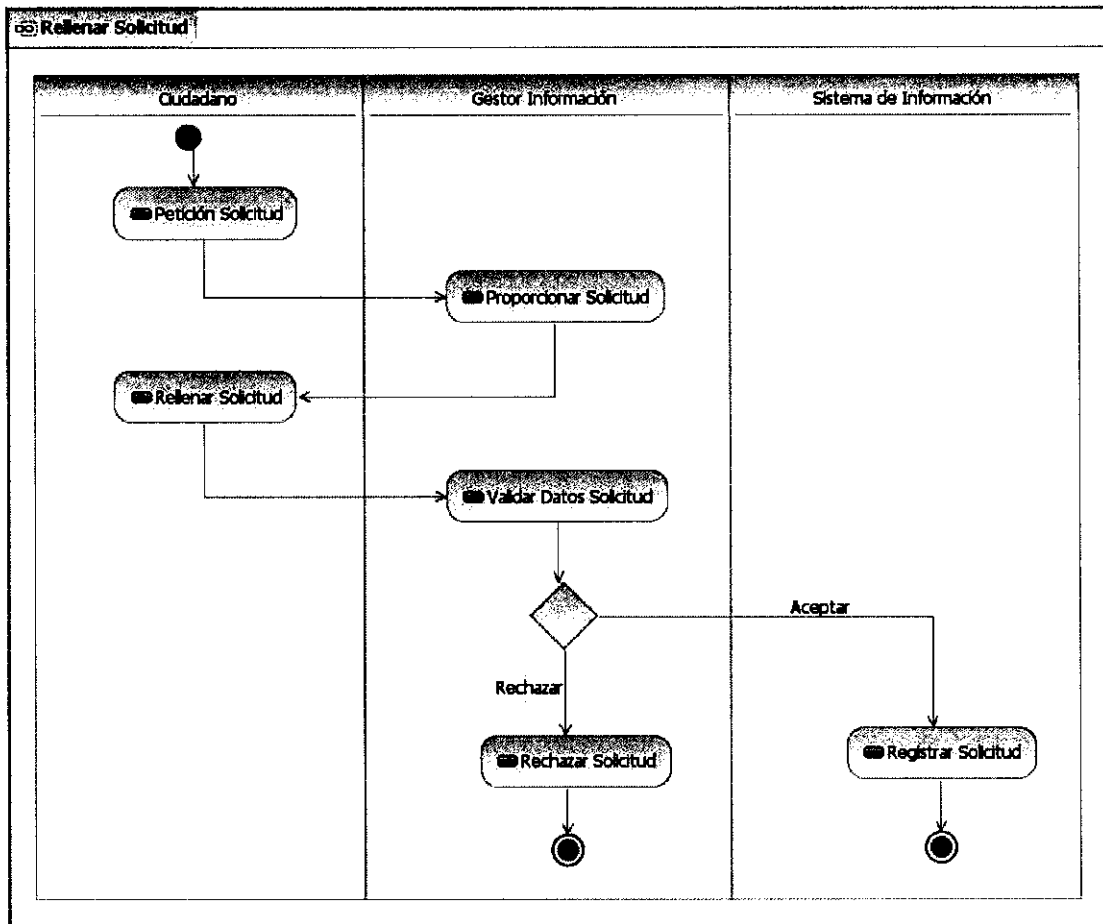


Figura 17. Business Use Case Realitzation Rellenar Solicitud

1. El Ciudadano solicita al SOC rellenar una Solicitud.
2. El Gestor de Información le proporciona el documento asociado a la Solicitud.
3. El Ciudadano rellena la Solicitud y se la entrega al Gestor de Información.
4. El Gestor de Información valida los datos aportados y en caso de que estos sean correctos, procede a registrar la Solicitud en el Sistema.
5. El Sistema registra la Solicitud aportada.

En caso de que se deseen examinar los diagramas de actividad del resto de procesos de negocio del *Servei d'Ocupació de Catalunya*, consultar en el anexo "**Software Architecture Document**" el subapartado *Business Use Case Realitzation* del apartado de *Business Model*.

5 Use Case Model

5.1 Introducción

A partir de la especificación de los requisitos funcionales del Sistema realizada previamente, se ha procedido a definir el Modelo de Casos de Uso del Sistema. Cabe destacar que el **Business Model** definido previamente **ha facilitado** considerablemente la definición de este Modelo de Casos de Uso. Si bien es cierto que la inclusión de los Terminales de Autoservicio modifica ligeramente gran parte de los servicios ofrecidos por el SOC, esta modificación se limita a la apertura de una nueva vía de comunicación entre el propio SOC y los Ciudadanos, una nueva vía cuyo principal cometido es automatizar una serie de tareas y funcionalidades. Así pues, esta automatización del proceso, como se podrá observar a lo largo de este apartado, aporta una serie de matices al desarrollo de las funcionalidades ofrecidas por los Puntos de Autoservicio, pero estas funcionalidades mantienen incuestionables similitudes con las ofrecidas actualmente. Por tanto, la automatización del proceso permitirá agilizar considerablemente la realización de una serie de tareas y servicios ofrecidos por el SOC, pero la esencia de éstos se mantendrá, con lo que el proceso de modelado realizado en el **Business Model** facilita enormemente la definición de los Casos de Uso del Sistema. No obstante, cabe destacar que ciertos servicios proporcionados por el SOC y modelados en el **Business Model** no forman parte de las funcionalidades ofrecidas en los Puntos de Autoservicio como es el caso de la publicación de nuevos Cursos de Formación y Ofertas de Empleo

Este Modelo de Casos de Uso permitirá obtener una descripción detallada de las diferentes funcionalidades del Sistema, así como la definición de la interacción de cada uno de los actores que utilizan el Sistema con las diferentes secuencias de acciones que pueden realizar. Esta descripción debe facilitar la comprensión de los requisitos funcionales que determinan el Sistema.

Con el fin de alcanzar esta serie de objetivos marcados, se ha procedido a definir:

- **Actores del Sistema**
- **Diagramas de Caso de Uso**
- **Descripción de las realizaciones**
- **Diagramas Actor-Sistema**

5.2 Actores

En primer lugar, se deben definir los diferentes actores que participan en los Casos de Uso del Sistema. Los Puntos de Autoservicio que se pretenden desarrollar estarán dirigidos y destinados a tres tipologías de usuarios bien diferenciadas, los **Ciudadanos**, los **Ciudadanos Inscritos** y los **Ciudadanos Registrados**. Estos 3 actores acaparan prácticamente la totalidad de funcionalidades y acciones que puede desarrollar el Sistema, exceptuando una serie de tareas de gestión y monitorización del propio Sistema llevadas a cabo por **personal del SOC**.

Las características asociadas a los roles **Ciudadano**, **Ciudadano Inscrito** y **Ciudadano Registrado** ya han sido comentadas a lo largo de la documentación.

Recordar, simplemente, que la diferencia entre estos 3 roles radica en las limitaciones de acceso a ciertas funcionalidades del Sistema definidas para los roles **Ciudadano** y **Ciudadano Inscrito**.

En la siguiente imagen se pueden observar las 4 tipologías de actores definidas, así como la relación que se establece entre ellos. Cabe destacar la presencia del actor **Trabajador SOC** que, como ya se ha comentado previamente, está asociado a una serie de tareas de gestión y administración de los Puntos de Autoservicio.

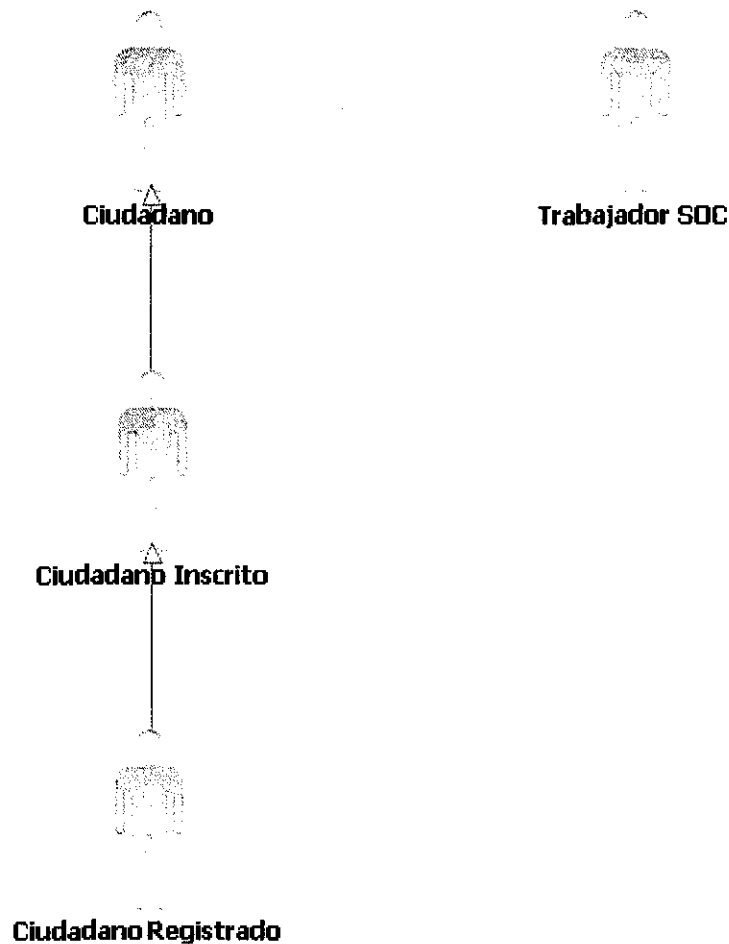


Figura 18. Actores del Sistema

5.3 Diagramas de Caso de Uso

En primer lugar, cabe realizar una descripción de cada uno de los Casos de Uso que conforman el Sistema de Puntos de Autoservicio. La definición de los Casos de Uso del Sistema está delimitada por la determinación de las funcionalidades del mismo realizada en la especificación de los requisitos funcionales. A partir de las funcionalidades definidas en dichos requisitos y, tratando de agrupar en un mismo Caso de Uso funcionalidades de una temática cercana, se han definido los siguientes Casos de Uso:

- **Gestión Cuenta y Demanda:** Conjunto de funcionalidades destinadas a gestionar toda la información asociada tanto a los datos personales del Ciudadano inscritos o registrados en el *Servei d'Ocupació de Catalunya*, como la información asociada a la demanda de empleo de dicho Ciudadano.
- **Gestión Información:** Conjunto de funcionalidades asociadas a procesos informativos establecidos entre el SOC y los Ciudadanos.
- **Gestión Formación:** Conjunto de funcionalidades que definen los diferentes procesos de interacción que se establecen entre los Ciudadanos y los Cursos de Formación ofrecidos por el SOC.
- **Gestión Ofertas:** Conjunto de funcionalidades que se definen los diferentes procesos de interacción que se establecen entre los Ciudadanos y las diferentes Ofertas de Empleo ofrecidas por el SOC.
- **Gestión Sesión:** Conjunto de funcionalidades destinadas a gestionar las sesiones de Sistema. Estas sesiones de Sistema son fruto del acceso a los Puntos de Autoservicio por parte de usuarios asociados a las tipologías **Ciudadano Inscrito** y **Ciudadano Registrado**. Cabe destacar que una de sus realizaciones, concretamente la asociada al inicio de una nueva sesión en el Sistema, es utilizada por otros Casos de Uso del Sistema. Esta utilización definirá una serie de **Relaciones de Inclusión** con el Caso de Uso Gestión Sesión.
- **Soporte y Atención al Cliente:** Conjunto de funcionalidades destinadas a solucionar y solventar las dudas que los usuarios pueden tener como consecuencia de su interacción con los Puntos de Autoservicio.
- **Gestión Punto Autoservicio:** Conjunto de funcionalidades asociadas a tareas de gestión y administración de los Puntos de Autoservicio.
- **Impresión Datos:** Funcionalidad asociada a la impresión de ciertos documentos a través de los Puntos de Autoservicio. Se trata de un Caso de Uso que no tiene sentido por sí sólo y cuya existencia viene determinada por la factorización de un comportamiento común en algunos de los Casos de Uso anteriores. Así pues, se definirá una **Relación de Extensión** entre el Caso de Uso **Impresión Datos** y otros Casos tales como **Gestión Información**. Esta relación de **Extensión**, se podrá observar en los Diagramas de Caso de Uso que se definirán a continuación.

En el apartado *"Descripción de las realizaciones"*, se podrán consultar los detalles de cada una de las realizaciones que conforman estos Casos de Uso definidos.

A continuación se procederá a definir las relaciones que se establecen entre los Actores del Sistema y los diferentes Casos de Uso mediante **Diagramas de Caso de Uso**. Dado que el objetivo de los **Diagramas de Caso de Uso** no es otra que determinar las relaciones que se establecen entre los Actores del Sistema y los Casos de Uso del mismo, se ha decidido mostrar por separado cada una de las relaciones Actor-Caso de Uso establecidas con el fin de facilitar la comprensión de dichas relaciones aportando una descripción y justificación de la misma. Estos **Diagramas de Caso de Uso** permitirán además observar de una manera clara y evidente a que funcionalidades del Sistema puede acceder cada tipo de usuario, ya que, como se ha comentado previamente, son estas limitaciones de acceso las que determinan la creación de diferentes tipologías de usuario para los ciudadanos de a pie.

5.3.1 Casos de Uso del Actor Ciudadano

5.3.1.1 Caso de Uso Gestión Cuenta y Demanda

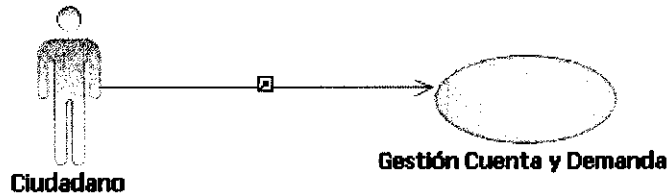


Figura 19. Caso de Uso Ciudadano-Gestión Cuenta y Demanda

El actor Ciudadano participa en una serie de realizaciones del Caso de Uso **Gestión Cuenta y Demanda**. Estas realizaciones son:

- **Inscripción Abreviada:** Esta realización permitirá al Ciudadano inscribirse en el Sistema, accediendo de este modo a prácticamente la totalidad de funcionalidades del mismo. A partir de esta realización, el Ciudadano pasará a ser un Ciudadano Inscrito.
- **Crear Cuenta:** Se trata de una realización opcional del Caso de Uso Gestión Cuenta y Demanda, ya que, como se ha comentado anteriormente en el apartado asociado a los requisitos del Sistema, es una propuesta de mejora de las funcionalidades que se espera que ofrezca el sistema. Esta realización permitirá al Ciudadano registrarse en el sistema, accediendo a la totalidad de funcionalidades del mismo, incluyendo los procesos de intermediación laboral. A partir de esta realización, el Ciudadano pasará a ser un Ciudadano Registrado.

5.3.1.2 Caso de Uso Gestión Información



Figura 20. Caso de Uso Ciudadano-Gestión Información

En la relación que se establece entre el Actor Ciudadano y el Caso de Uso **Gestión Información** se puede observar la primera relación de extensión asociada al Caso de Uso **Impresión Datos**. Así pues, el usuario de tipo Ciudadano participa en las siguientes realizaciones del Caso de Uso **Gestión Información**.

- **Acceso Información:** Esta realización permitirá al Ciudadano informarse sobre los servicios y prestaciones que ofrece el SOC, así como la documentación exigida en cada uno de los trámites que debe realizar a través del SOC. Dado que el Ciudadano puede realizar la impresión de esta información, se ha definido una relación de Extensión con el Caso de Uso **Impresión Datos**.

- **Rellenar Solicitud:** Con esta realización, un Ciudadano podrá rellenar y enviar al SOC, mediante el punto de Autoservicio, una serie de solicitudes tales como 'Prestación Contributiva' o 'Subsidio de paro'.

5.3.1.3 Caso de Uso Gestión Formación

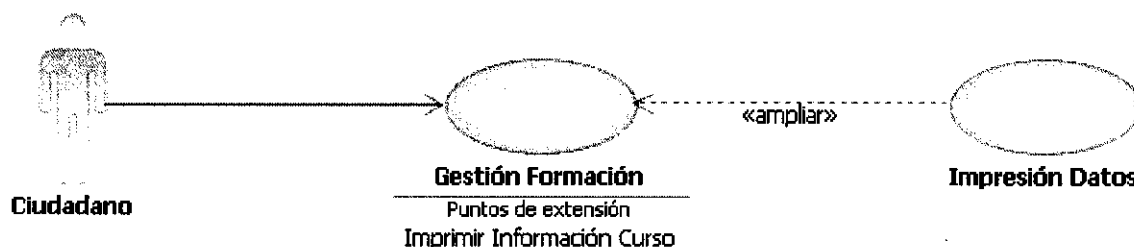


Figura 21. Caso de Uso Ciudadano-Gestión Formación

El Actor identificado como Ciudadano participa en una de las realizaciones que conforman el Caso de Uso **Gestión Formación**.

- **Buscar Cursos:** Esta realización permitirá al Ciudadano consultar la información de una serie de cursos de formación que se ofrecen a través del Punto de Autoservicio. Dado que el Ciudadano puede realizar la impresión de los datos asociados al Curso de Formación, se ha definido una realización de Extensión con el Caso de Uso **Impresión Datos**.

5.3.1.4 Caso de Uso Gestión Ofertas

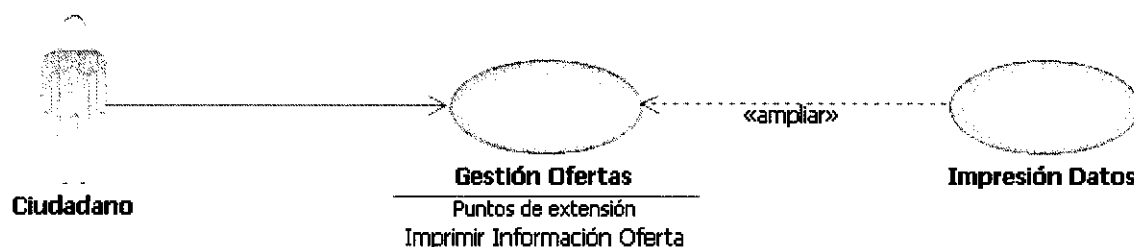


Figura 22. Caso de Uso Ciudadano-Gestión Ofertas

El usuario de tipo Ciudadano participa en una de las realizaciones que forman parte del Caso de Uso **Gestión Ofertas**.

- **Buscar Ofertas:** Esta realización permite que un Ciudadano consulte los detalles de un conjunto de ofertas de empleo a través del Punto de Autoservicio. Considerando que el Ciudadano puede realizar la impresión de los datos de una Oferta de Empleo seleccionada, se ha definido una relación de Extensión con el Caso de Uso **Impresión Datos**.

5.3.1.5 Caso de Uso Soporte y Atención al Cliente

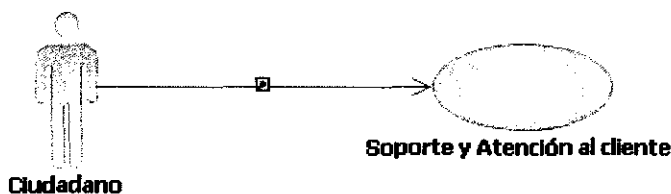


Figura 23. Caso de Uso Ciudadano-Soporte

El actor asociado a la tipología de Ciudadano participa también en la única realización que conforma el Caso de Uso **Soporte y Atención al Cliente**.

- **Petición Soporte:** Esta realización esta encaminada a solucionar cualquier duda o problema que pueda surgirle a un ciudadano con su interacción con el sistema.

5.3.2 Casos de Uso del Actor Ciudadano Inscrito

5.3.2.1 Caso de Uso Gestión Cuenta y Demanda

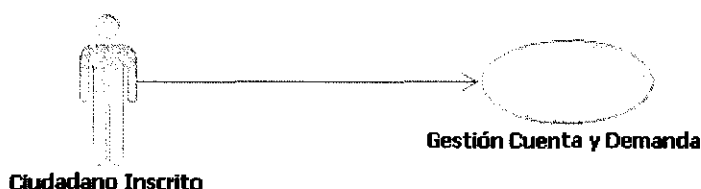


Figura 24. Caso de Uso Ciudadano Inscrito-Gestión Cuenta y Demanda

El actor Ciudadano Inscrito participa en una serie de realizaciones que conforman el Caso de Uso **Gestión Cuenta y Demanda**. El acceso a estas funcionalidades se ha producido gracias a su ingreso en el sistema mediante la funcionalidad **Inscripción Abreviada**.

- **Eliminar Cuenta:** Se trata de una realización opcional del Caso de Uso Gestión Cuenta y Demanda, ya que, como se ha comentado anteriormente en el apartado asociado a los requisitos del Sistema, es una propuesta de mejora de las funcionalidades que se espera que ofrezca el Sistema. Esta realización está destinada a eliminar del Sistema todos los datos asociados al ciudadano. Tras esta eliminación, el Ciudadano Inscrito volverá a convertirse en un actor de tipo Ciudadano.
- **Actualizar Perfil:** Esta realización permitirá al Ciudadano Inscrito modificar sus datos almacenados en el Sistema.
- **Activación Demanda:** Se trata de una realización asociada a la demanda de empleo por parte del ciudadano. Esta realización le permite activar la demanda pasando a tener ésta un estado de 'Alta'.
- **Renovación Demanda:** Con esta realización y utilizando una clave de acceso proporcionada por el Sistema, un Ciudadano Inscrito podrá renovar, dentro del periodo establecido, su petición de demanda de empleo.

5.3.2.2 Caso de Uso Gestión Información

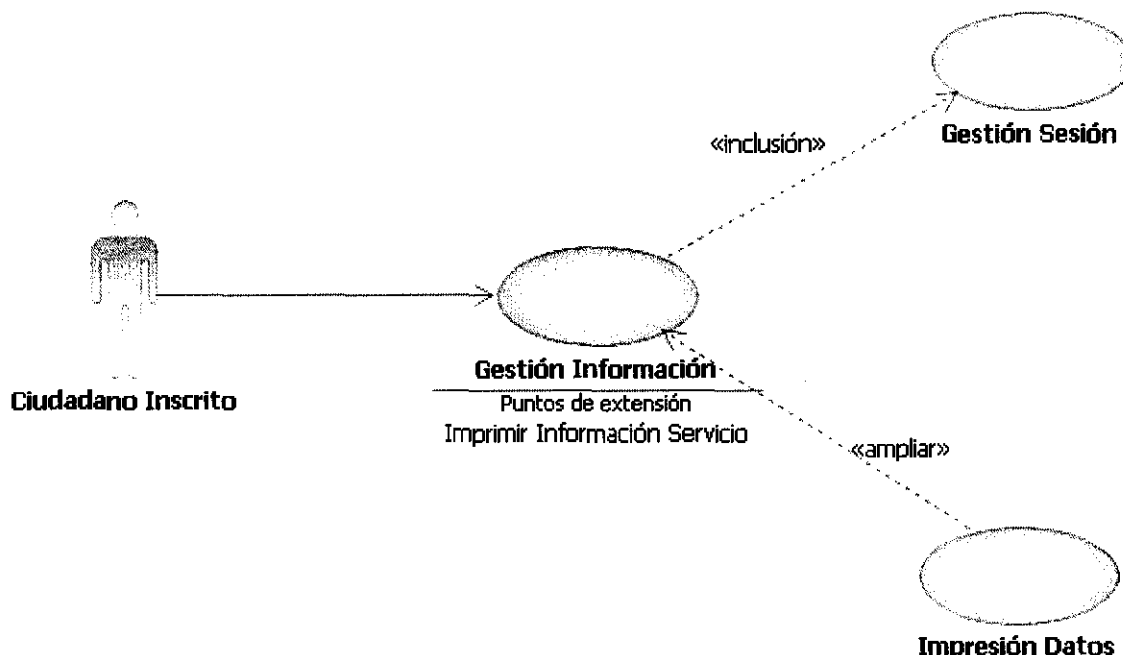


Figura 25. Caso de Uso Ciudadano Inscrito-Gestión Información

En la imagen anterior, se puede observar que en la interacción del Actor Ciudadano Inscrito con el Caso de Uso **Gestión Información** se establecen una relación de inclusión con el Caso de Uso **Gestión Sesión** y una de extensión con **Impresión Datos**. La relación con este último, como ya se ha comentado previamente, viene determinada por el hecho de que los usuarios pueden imprimir una serie de documentos mostrados en los Puntos de Autoservicio. Así pues, cabe destacar la **Relación de Inclusión** que se establece con el Caso de Uso **Gestión Sesión**. Como se puede observar en el documento de especificaciones técnicas aportado por el SOC y adjuntado en esta documentación, existen una serie de funcionalidades que varían su comportamiento en función de si el usuario que realiza dicha funcionalidad ha iniciado o no una sesión en el Sistema. Esta situación sucede por ejemplo en la realización **Rellenar Solicitud** del Caso de Uso **Gestión Información**. Dicha variación del comportamiento comporta la **Relación de Inclusión** comentada anteriormente. Así pues, el actor Ciudadano Inscrito tiene una participación importante en las funcionalidades que forman el Caso de Uso **Gestión Información**. Las realizaciones en las que participa de dicho caso son las siguientes:

- **Petición Informe:** Mediante esta realización, un Ciudadano Inscrito podrá solicitar al sistema la impresión de una serie de informes tales como 'Informe de Prestaciones' o 'DARDO' relacionados con su demanda de empleo. Esta realización es la que determina la extensión del Caso de Uso **Impresión Datos**, ya que el Sistema ofrecerá al Ciudadano Inscrito la posibilidad de imprimir los datos de sus Informes.
- **Rellenar solicitud:** A esta realización ya tenía acceso un ciudadano sin necesidad de inscribirse o registrarse en el sistema, pero tras hacerlo, a la hora de rellenar una solicitud el sistema se encargará de agilizar este proceso evitando que el usuario tenga que proporcionar datos ya conocidos. Esta realización es la que determina la inclusión del Caso de Uso **Gestión Sesión**,

ya que el Ciudadano Inscrito inicia una sesión en el sistema para identificarse y que de este modo el Sistema rellene una serie de datos ya conocidos.

5.3.2.3 Caso de Uso Gestión Formación

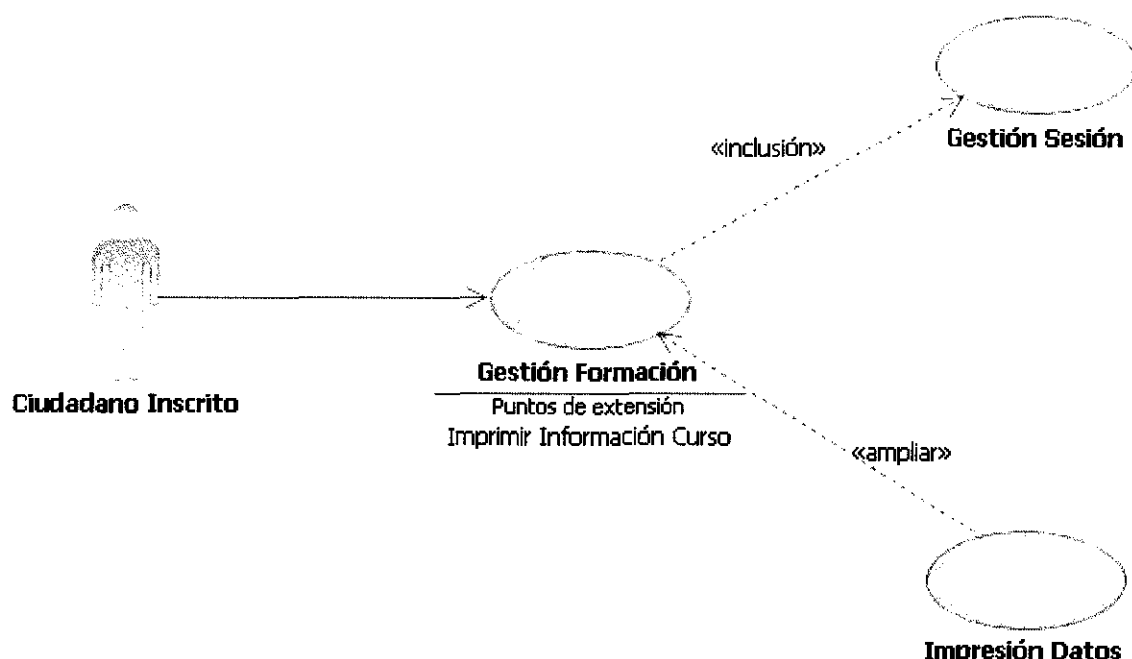


Figura 26. Caso de Uso Ciudadano Inscrito-Gestión Formación

Al igual que en el Caso de Uso **Gestión Información**, debido a la interacción del Actor Ciudadano Inscrito con el Caso de Uso **Gestión Formación** se establece una relación de inclusión con el Caso de Uso **Gestión Sesión** y una de extensión con **Impresión Datos**. La justificación de estas relaciones es la misma que para el Caso de Uso **Gestión Información**. Así pues, con respecto al Caso de Uso **Gestión Formación**, cabe destacar que un Ciudadano Inscrito puede participar en las funcionalidades de Buscar Cursos e Inscripción Curso gracias a la creación de su cuenta en el sistema.

- **Buscar Cursos:** Esta realización permite que un Ciudadano Inscrito consulte los detalles de un conjunto de cursos de formación a través del Punto de Autoservicio. Esta realización ya estaba disponible para los usuarios de tipo Ciudadano, pero si el usuario inicia una sesión en el sistema, de ahí la inclusión del Caso de Uso **Gestión Sesión**, el Sistema mostrará al Ciudadano Inscrito los cursos de formación más acordes a su perfil. Por otra parte, dado que Ciudadano Inscrito puede realizar la impresión de los datos asociados a uno de los Cursos de Formación buscados, se ha definido una relación de extensión con el Caso de Uso **Impresión Datos**.
- **Inscripción Curso:** Esta realización del Caso de Uso **Gestión Formación** permite que un usuario pueda inscribirse en uno de los Cursos de Formación ofrecidos por el sistema con el fin de ampliar sus conocimientos y aptitudes.

5.3.2.4 Caso de Uso Gestión Ofertas

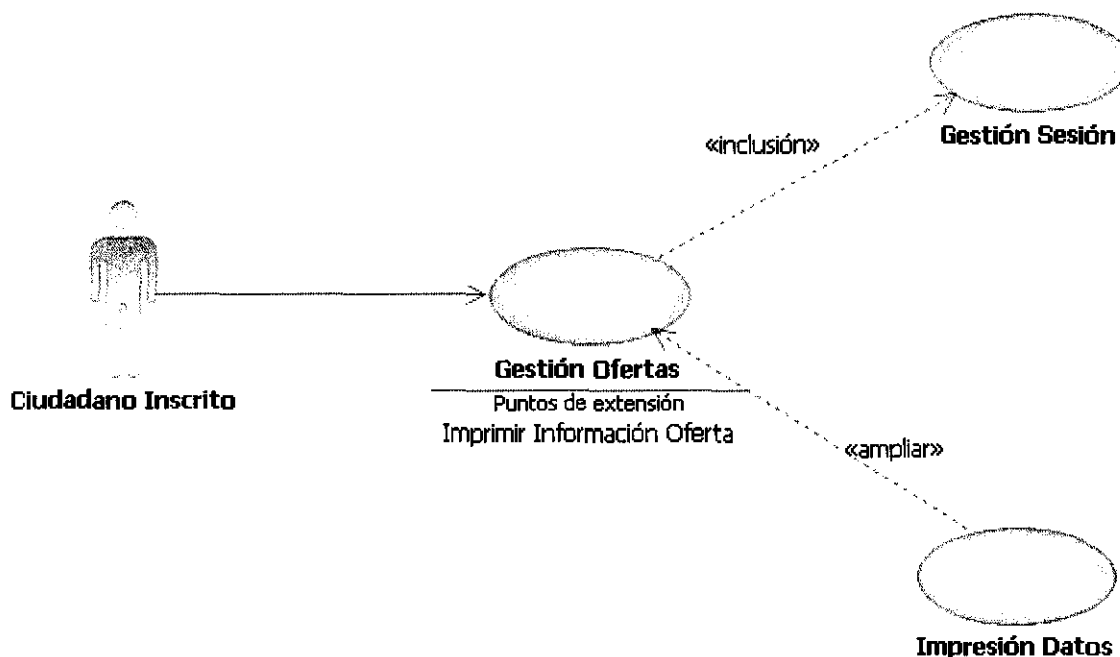


Figura 27. Caso de Uso Ciudadano Inscrito-Gestión Ofertas

De nuevo nos encontramos con un diagrama muy similar al mostrado para el Caso de Uso **Gestión Formación**. Debido a la interacción del Actor Ciudadano Inscrito con el Caso de Uso **Gestión Ofertas**, se establecen una relación de inclusión con el Casos de Uso **Gestión Sesión** y una de extensión con el Caso **Impresión Datos**. La relación de extensión con el Caso de Uso **Impresión Datos** viene determinada por el hecho que el Sistema ofrece al Ciudadano imprimir una serie de datos de las Ofertas de Empleo. En cambio, la relación de inclusión con el Caso de Uso **Gestión Sesión** viene determinada, al igual que en los Casos de Uso anteriores, por una variación del comportamiento de la realización **Buscar Ofertas** cuando el usuario que lleva dicha realización ha iniciado previamente una sesión en el Sistema. Así pues, el usuario de tipo 'Ciudadano Inscrito' participa en una de las realizaciones que forman parte del Caso de Uso Gestión Ofertas.

- **Buscar Ofertas:** Esta realización permite que un Ciudadano Inscrito consulte los detalles de un conjunto de ofertas de empleo a través del Punto de Autoservicio. Esta realización ya estaba disponible para los usuarios de tipo Ciudadano, pero si el usuario inicia una sesión en el sistema, de ahí la inclusión del Caso de Uso **Gestión Sesión**, el Sistema mostrará al Ciudadano Inscrito las ofertas de empleo más acordes a su perfil. Por otra parte, dado que el Ciudadano Inscrito puede realizar la impresión de los datos asociados a una de las Ofertas buscadas, se ha definido una relación de extensión con el Caso de Uso **Impresión Datos**.

5.3.2.5 Caso de Uso Gestión Sesión



Tras la creación de una cuenta de usuario, el Sistema dispone de una serie de funcionalidades destinadas a gestionar las sesiones de un Ciudadano Inscrito en el Sistema. Las realizaciones de este Caso de Uso están relacionadas con el inicio y el fin de estas sesiones.

- **Login:** Esta realización permite a un Ciudadano Inscrito iniciar una sesión en el Sistema, permitiéndole acceder a una serie de funcionalidades que están disponibles para aquellos usuarios que están inscritos o registrados en el Sistema.
- **Logout:** Esta realización permite a un Ciudadano Inscrito finalizar y cerrar su sesión en el Sistema.

5.3.3 Casos de Uso del Actor Ciudadano Registrado

5.3.3.1 Caso de Uso Gestión Ofertas



Como ya se ha comentado a lo largo de esta documentación, los procesos de intermediación laboral están reservados para Ciudadanos que han creado una cuenta en el Sistema, es decir, Ciudadanos Registrados. Así pues, el actor Ciudadano Registrado participa en una de las realizaciones asociadas al Caso de Uso **Gestión Ofertas**. El acceso a esta funcionalidad es posible gracias a que el ciudadano dispone de una cuenta en el sistema creada mediante la realización Crear Cuenta del Caso de Uso **Gestión Cuenta y Demanda**.

- **Inscripción Oferta:** Esta realización constituye seguramente la razón de ser del Sistema que se pretende desarrollar, los procesos de intermediación laboral. Gracias a esta realización, un ciudadano podrá inscribirse en una de las ofertas de trabajo publicadas en el Sistema y ser por tanto candidato a ocupar el puesto de trabajo ofertado.

5.3.4 Casos de Uso del Actor Trabajador SOC

Finalmente, se describirán aquellos Casos de Uso que tienen como protagonista al Trabajador del SOC, alejándose del patrón seguido hasta el momento donde los ciudadanos eran los que interactuaban con el Sistema.

5.3.4.1 Caso de Uso Gestión Punto Autoservicio

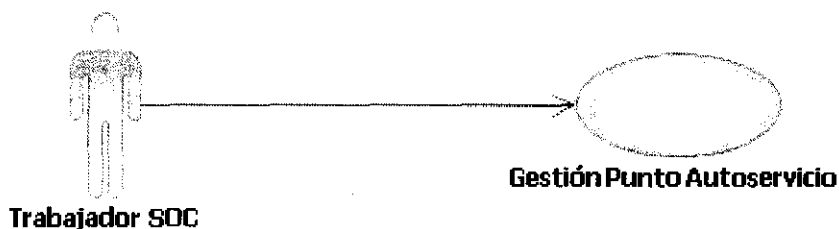


Figura 30. Caso de Uso Trabajador SOC-GEstion Punto Autoservicio

El actor Trabajador SOC protagoniza todas las realizaciones asociadas al Caso de Uso **Gestión Punto Autoservicio**, realizaciones encaminadas a gestionar y monitorizar el funcionamiento de dichos puntos. Cabe destacar que estas realizaciones no se realizan a través de los Puntos de Autoservicio, sino que el Trabajador del SOC debe realizar un **acceso remoto** al Sistema para efectuar dichas tareas de gestión y administración.

- **Activar Servicio:** Esta realización describe la activación de uno de los servicios ofrecidos en los Puntos de Autoservicio para un conjunto de estos. Así pues, el Trabajador del SOC deberá especificar tanto el servicio desea activar como los Puntos de Autoservicio que experimentarán dicha activación.
- **Desactivar Servicio:** Se trata simplemente de la realización opuesta a la anterior. Por tanto, esta realización describe la desactivación de uno de los servicios proporcionados por los Puntos de Autoservicio. Del mismo modo que sucede con la activación, el Trabajador del SOC deberá indicar el servicio que desea desactivar y los Puntos de Autoservicio que sufrirán dicha desactivación.

5.4 Descripción de las Realizaciones

La **Descripción de las Realizaciones** que determinan los Casos de Uso del Sistema consiste en una descripción textual de las diferentes interacciones que se producen entre los usuarios y el Sistema a lo largo de una realización.

Para cada realización se realizará una descripción de tanto el escenario principal como cada uno de los escenarios alternativos asociados a una realización. Además, se indicará:

- **Actores** que participan en la realización.
- **Precondiciones** que deben cumplirse para que se inicie la realización.
- **Postcondiciones** que deben cumplirse tras completar la realización, indicando de este modo en que estado queda el Sistema tras finalizar dicha realización.

Cabe destacar que a la hora de diseñar estas realizaciones se han tenido muy presentes los requisitos no funcionales definidos previamente. Así pues, y dada la importancia que la usabilidad tiene en los Puntos de Autoservicio, la creación de pantallas muy ligeras ha sido muy valorada a la hora de definir cada una de las

realizaciones que conforman los Casos de Uso. Estas consideraciones han llevado a dividir gran parte de los formularios asociados a estas realizaciones en diferentes páginas, dando lugar a una estructura en forma de *wizard*. Esta estructura se representará en la **Descripción de las Realizaciones** indicando que una serie de pasos o interacciones se repiten de manera iterativa.

Por otra parte, cabe destacar que en una serie de realizaciones se definen relaciones de inclusión y extensión con otras realizaciones. Estas relaciones de inclusión se representan simplemente indicando el nombre de la realización que es incluida acompañada del texto "Realización", es decir, un ejemplo de inclusión de la realización **Actualizar Perfil** sería "Realización: Actualizar Perfil". En el caso de las relaciones de extensión, una vez descritos los flujos principales y alternativos de la realización, se especificarán los **puntos de extensión** de dicha realización indicando para cada uno de ellos la condición de activación.

Finalmente, cabe reseñar que como se ha comentado en diversas ocasiones a lo largo de esta documentación, existen una serie de funcionalidades que varían su comportamiento en función del tipo de usuario que inicia dicha realización. Es el caso de las realizaciones **Rellenar Solicitud** del Caso de Uso **Gestión Información**, **Buscar Cursos** del Caso de Uso **Gestión Formación** y **Buscar Ofertas** del Caso de Uso **Gestión Ofertas**. Para representar dicha variación de comportamiento se ha utilizado una única realización, pero definiendo escenarios o flujos alternativos para representar dicha variación.

Así pues, a continuación se incluye una pequeña muestra de la descripción textual de las realizaciones que componen los Casos de Uso del Sistema con el fin mostrar un ejemplo de lo descrito anteriormente. Cabe destacar que el resto de las descripciones de las realizaciones se pueden consultar en el anexo "*Descripción de las Realizaciones de los Casos de Uso*" adjuntado a esta documentación.

El ejemplo seleccionado corresponde a la realización **Buscar Cursos** del Caso de Uso **Gestión Formación**. Se ha seleccionado esta realización ya que en ella se incluyen cada uno de los aspectos destacados anteriormente como son:

- Escenario principal y flujos alternativos.
- Puntos de Extensión.
- Flujo alternativo para representar las variaciones de comportamiento determinadas por el tipo de actor que inicia la realización.

Realización: Buscar Cursos

Actores: Ciudadano, Ciudadano Inscrito

{Pre: La persona que inicia la realización del Caso de Uso debe estar consultando el área de Gestión de Formación.}

{Post: El sistema muestra al ciudadano una serie de cursos de formación a los que el Ciudadano puede inscribirse.}

Flujo principal:

Ciudadano	Sistema
1. El Ciudadano decide realizar una búsqueda de cursos de formación.	
	2. El Sistema le pide que defina una serie de restricciones o filtros de búsqueda.
3. El Sistema muestra la lista de cursos de	

formación que se ajustan al filtro especificado.	
	4. El Sistema muestra la lista de Cursos de Formación que se ajustan al filtro especificado.
5. El Ciudadano selecciona uno de los cursos mostrados.	
	6. El Sistema muestra al Ciudadano toda la información asociada al curso.
7. El Ciudadano retorna al Listado de Cursos para consultar otros Cursos de Formación. REPETIR PASOS 4,5,6,7,8 HASTA QUE EL CIUDADANO HAYA CONSULTADO TODOS LOS CURSOS QUE LE INTERESAN DEL LISTADO MOSTRADO	

Flujo alternativo 1: El Ciudadano inicia una sesión en el sistema para realizar la búsqueda de cursos.

Ciudadano Inscrito	Sistema
1. <u>Realización: Login</u>	
2. El Ciudadano decide realizar una búsqueda de cursos de formación.	
	3. El Sistema le pide que defina una serie de restricciones o filtros de búsqueda.
4. El Ciudadano define un filtro de búsqueda.	
	5. El Sistema muestra la lista de cursos de formación que se ajustan al filtro especificado e indica aquellos cursos que se ajustan más a su perfil. Volver al paso 5 del flujo principal

Flujo alternativo 2: No existe ningún Curso de Formación que cumpla con los parámetros de búsqueda especificados (4).

Ciudadano	Sistema
	1. El Sistema indica al ciudadano que no existe ningún Curso de Formación que cumpla con los parámetros de búsqueda especificados y le ofrece la posibilidad de abandonar la búsqueda o realizar una nueva.

Flujo alternativo 3: El Ciudadano decide no consultar los detalles de ningún curso en concreto (5).

Acaba la realización del Caso de Uso.

Punto de Extensión: Imprimir Información Curso (6).

- **Condición de Activación:** El Ciudadano decide imprimir la información del Curso de Formación.

5.5 Diagramas Actor-Sistema

Los Diagramas Actor-Sistema tienen como objetivo representar gráficamente la interacción que se produce entre el usuario y el Sistema de Información en cada una de las realizaciones que componen los Casos de Uso del Sistema. Esta representación de las interacciones debe permitir observar con claridad:

- Los eventos externos generados por los usuarios o actores.
- El orden en que se produce la secuencia de interacciones y eventos.
- El conjunto de eventos internos (operaciones) que realiza el Sistema para responder a los eventos externos.



Así pues, para cada una de las realizaciones de los Casos de Uso del Sistema definidas en el apartado “*Descripción de las Realizaciones*”, se ha procedido a definir el Diagrama Actor-Sistema correspondiente.

No obstante, cabe destacar que algunas realizaciones han requerido un tratamiento especial. Este es el caso de las realizaciones **Rellenar Solicitud**, **Buscar Cursos** y **Buscar Ofertas**. Estas tres realizaciones tienen en común que su comportamiento varía en función de la tipología del usuario que la lleva a cabo. Así pues, y a consecuencia de esta variación, se ha procedido a definir dos Diagramas Actor-Sistema diferentes, uno por cada tipología de usuario, para cada una de estas realizaciones. Cabe destacar que en la descripción textual de estas realizaciones definida en el apartado “*Descripción de las Realizaciones*”, esta variación de comportamiento se ha tratado simplemente definiendo un escenario o flujo alternativo que representase dicha variación. Sin embargo, considerando que los **Diagramas Actor-Sistema** pretenden describir gráficamente la interacción que se produce entre el usuario y el Sistema a lo largo de una realización, la variación de la tipología de este usuario obliga a definir un diagrama para cada tipo de actor.

A continuación, pueden observarse una pequeña muestra de los **Diagramas Actor-Sistema** definidos.

Realización Buscar Cursos del Caso de Uso Gestión Formación

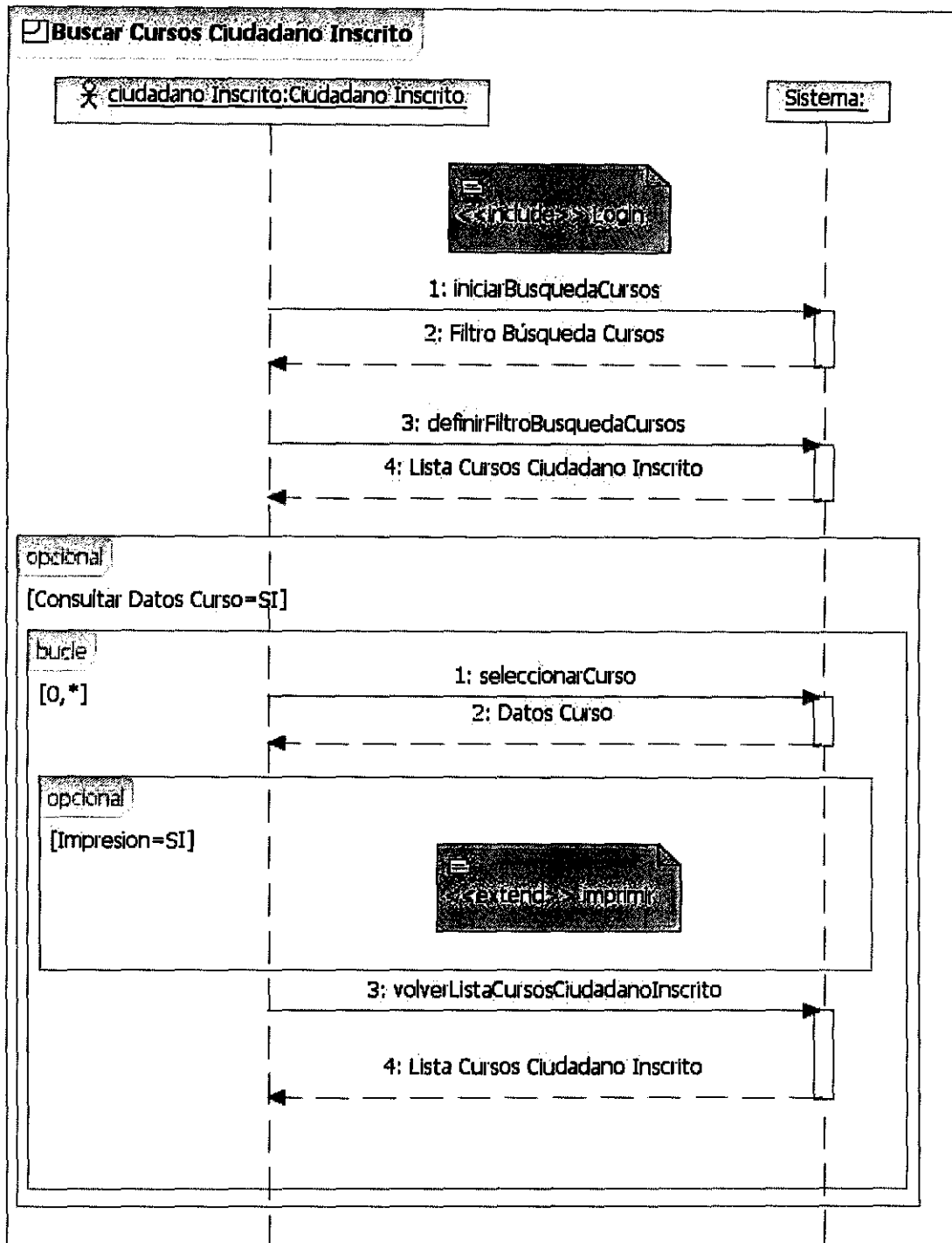


Figura 31. Diagrama Actor-Sistema Buscar Cursos

La imagen anterior muestra el Diagrama Actor-Sistema de la realización **Buscar Cursos** del Caso de Uso **Gestión Formación**. Como se ha comentado previamente, para esta realización se han realizado dos diagramas diferentes, uno para el actor Ciudadano y otro para el actor Ciudadano Inscrito. La imagen mostrada corresponde precisamente a este último. En apartados anteriores de esta documentación, se ha comentado que la realización **Buscar Cursos** determina las relaciones de inclusión y extensión del Caso de Uso **Gestión Formación** con los Casos de Uso **Gestión Sesión** e **Impresión Datos** respectivamente. Estas relaciones se representan en el diagrama mediante un elemento de tipo "Comentario". Así pues, en la imagen anterior podemos observar la relación de inclusión con la realización

Login del Caso de Uso **Gestión Sesión**, así como la relación de extensión con la realización **Imprimir** del Caso de Uso **Impresión Datos**.

Realización Crear Cuenta del Caso de Uso Gestión Cuenta y Demanda

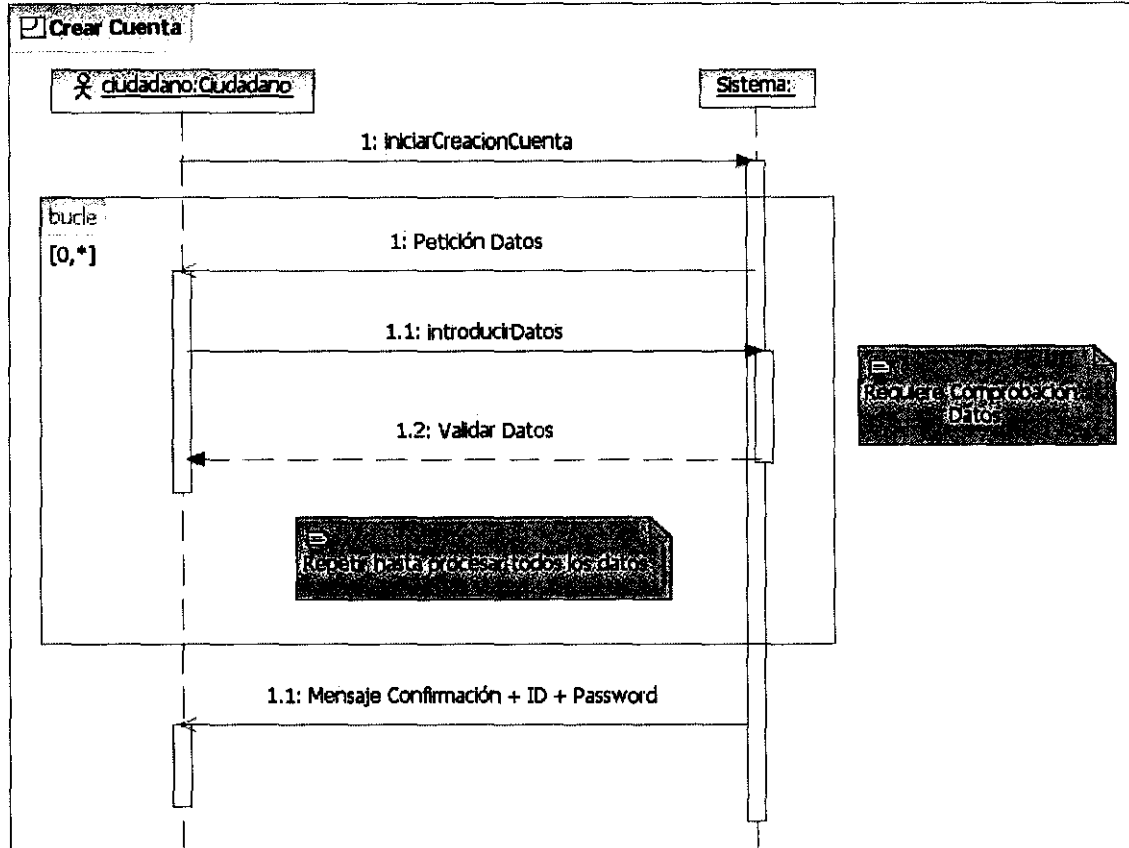


Figura 32. Diagrama Actor-Sistema Crear Cuenta

En la imagen anterior, podemos observar el Diagrama Actor-Sistema de la realización Crear Cuenta del Caso de Uso Gestión Cuenta y Demanda. Cabe destacar la inclusión del bucle que pretende representar la estructura en forma de *Wizard* que define dicha realización tal y como ya se ha comentado previamente en el apartado "Descripción de las Realizaciones". Por otra parte, en la descripción textual de esta realización se han definido una serie de flujos alternativos debido a la introducción de datos erróneos por parte del usuario. Esta serie de comprobaciones por parte del Sistema con el consiguiente mensaje de error se han representado en el Diagrama Actor-Sistema mediante inclusión del comentario "Requiere Comprobación Datos".

En caso de que se deseen examinar los **Diagramas Actor-Sistema** del resto de realizaciones que conforman el **Use Case Model** del Sistema, consultar en el anexo "**Software Architecture Document**" el subapartado *Casos de Uso y Realizaciones* del apartado *Analysis Model*.

6 Storyboards

El diseño de la Interfaz de Usuario, es decir, el medio que permite a un usuario comunicarse e interaccionar con el Sistema, acostumbra a jugar un papel fundamental en la aceptación del Sistema por parte de sus usuarios. Esta importancia se incrementa en sistemas como los Puntos de Autoservicio del SOC, dado que los usuarios de estos terminales no están acostumbrados a interaccionar con sistemas similares. Así pues, el diseño de esta interfaz deberá ajustarse **a las necesidades de este tipo de usuarios** con el objetivo de que su interacción con el Sistema se produzca de una manera cómoda y eficaz.





En la mayoría de las ocasiones esta definición de la interficie gráfica conocida como *StoryBoards* se limita a plasmar sobre papel el diseño gráfico de cada una de las pantallas que conforman el Sistema. Sin embargo, para este proyecto de análisis y diseño de los Puntos de Autoservicio se ha considerado que, dado que gran parte de las posibilidades de éxito del proyecto radican en un adecuado diseño de la interficie gráfica del Sistema, este diseño no podía limitarse a plasmar sobre el papel el aspecto gráfico de cada una de las pantallas que conforman la aplicación. Así pues, se procedió a realizar una **implementación física** de dichas pantallas con el fin de que la evaluación del correcto diseño de las mismas no se limite a analizar su aspecto y diseño gráfico, sino que la posibilidad de interactuar con dichas pantallas gracias a la implementación física de las mismas permita evaluar aspectos tan importantes en el Sistema de Puntos de Autoservicio como son la **usabilidad y accesibilidad**.

Para realizar esta implementación física de los *StoryBoards* se ha optado, siguiendo las directrices impuestas en el documento de preinscripciones técnicas aportado por el SOC, por utilizar lenguajes de programación basados en la **tecnología J2EE** tales como **JSF, JSP y JavaScript**.

A pesar de que la implementación física de la interficie gráfica carece en todo momento de una Capa de Negocio que sustente las interacciones que el usuario realiza con el Sistema, dicha implementación permite acceder a cada una de las funcionalidades del Sistema, simulando incluso el inicio de una sesión en el mismo. Esta simulación del inicio de sesión permitirá al usuario acceder a las funcionalidades reservadas a Ciudadanos Inscritos o Registrados. Así pues, a pesar de carecer de una Capa de Negocio, la implementación física de estos *StoryBoards* se ha realizado de manera que pueda considerarse una representación fidedigna de la interficie del Sistema una vez finalizada la implementación física del mismo.

A continuación se muestran una serie de ejemplos de los *Storyboards* definidos. Cabe destacar que el **anexo "Interfaz de Usuario"** adjuntado a esta documentación contiene una muestra más amplia de las pantallas que conforman esta interfaz, incluyendo tanto una exhaustiva demostración del cumplimiento de los requisitos no funcionales definidos en los *Storyboards* creados, como los *Navigation Maps* que determinan el funcionamiento de esta interfaz de usuario.

SOC



En las imágenes anteriores, resulta sencillo observar que la interfaz de usuario definida está formada por **pantallas ligeras** que utilizan **iconos de reconocimiento internacional**, ajustándose en todo momento a las características de los usuarios que interaccionarán con dicha interfaz. Cada uno de estos detalles se describe con detenimiento en el anexo indicado anteriormente.

7 Analysis Model

Estamos ante uno de los elementos fundamentales del proyecto de análisis y diseño de los Puntos de Autoservicio del SOC. El Modelo de Análisis describe de manera detallada la estructura del Sistema que se pretende desarrollar mediante el **Modelo Conceptual** o Diagrama de Clases del Sistema y una serie de diagramas de secuencia. Este conjunto de diagramas pretenden realizar una descripción lógica de la implementación de los requisitos funcionales definidos previamente.

El **Modelo Conceptual** describirá las diferentes clases participantes en el Sistema definiendo de este modo el aspecto estático del mismo. Por otra parte, los diagramas de secuencia mencionados pretenden describir de manera detallada el flujo de interacciones que se realizan entre las diferentes clases del Sistema en cada una de las realizaciones de los Casos de Uso definidos.

7.1 Diagrama de Clases

A continuación se muestra el **Diagrama de Clases** o **Modelo Conceptual** del Sistema. Cabe destacar que, pese a que este diagrama pertenece al Modelo de Análisis del Sistema, modelos posteriores tales como el Modelo de Diseño varían la estructura de dicho diagrama debido a la normalización del mismo o a la aplicación de patrones de diseño. No obstante, la imagen que se muestra a continuación corresponde al Diagrama de Clases definitivo, es decir, el resultante de aplicar todas las modificaciones asociadas a modelos posteriores. Como se puede observar, se muestran por separado las Restricciones de Integridad del Diagrama con el objetivo de facilitar la lectura de estas.

RESTRICCIONES DE INTEGRIDAD:

1. Ciudadano Inscrito no puede tener asociadas 2 Inscripciones Servicio para el mismo servicio tales que la fecha de inicio de una inscripción esté comprendida entre la fecha de inicio y la fecha de fin de la otra inscripción.
2. Para un Ciudadano Inscrito, no pueden existir 2 solicitudes del mismo Tipo tales que el estado de ambas sea "En Estudio".
3. Para un Ciudadano Inscrito, no pueden existir 2 informes del mismo Tipo.
4. Un Ciudadano Inscrito no puede percibir 2 veces la misma prestación a pesar que esta pertenezca a Servicios diferentes a los que está inscrito el Ciudadano.
5. El número de datos de un Informe debe ser \leq número de campos del Tipo de ese Informe.
6. El número de datos de una Solicitud debe ser \leq número de campos del Tipo de esa Solicitud.
7. Fecha Inscripción Curso \geq Fecha Inscripción en el Sistema del Ciudadano que se inscribe en el Curso.
8. Fecha Inscripción Oferta \geq Fecha Inscripción en el Sistema del Ciudadano que se inscribe en la Oferta.
9. La Fecha de un Informe \geq Fecha Inscripción en el Sistema del Ciudadano al que pertenece ese Informe.
10. Para todas las Solicitudes asociadas a un Ciudadano Inscrito, la Fecha de la Solicitud debe ser \geq Fecha Inscripción del Ciudadano en el Sistema.
11. Para todas las Inscripciones de Servicios asociadas a un Ciudadano, la Fecha de esa Inscripción debe ser \geq Fecha Inscripción del Ciudadano en el Sistema.
12. No puede haber 2 instancias de la clase Inscripción Curso para el mismo Ciudadano Inscrito y el mismo Curso Formación.
13. No puede haber 2 instancias de la clase Inscripción Oferta para el mismo Ciudadano Inscrito y la misma Oferta Empleo.
14. No puede haber 2 instancias de la clase Estado Funcionalidad para el mismo Punto Autoservicio y la misma Funcionalidad.
15. No puede haber 2 instancias de la clase Inscripción Servicio para el mismo Ciudadano Inscrito, el mismo Servicio y la misma Fecha.
16. No puede haber 2 instancias de la clase Perfil Demanda para el mismo Ciudadano Inscrito y la misma Demanda.

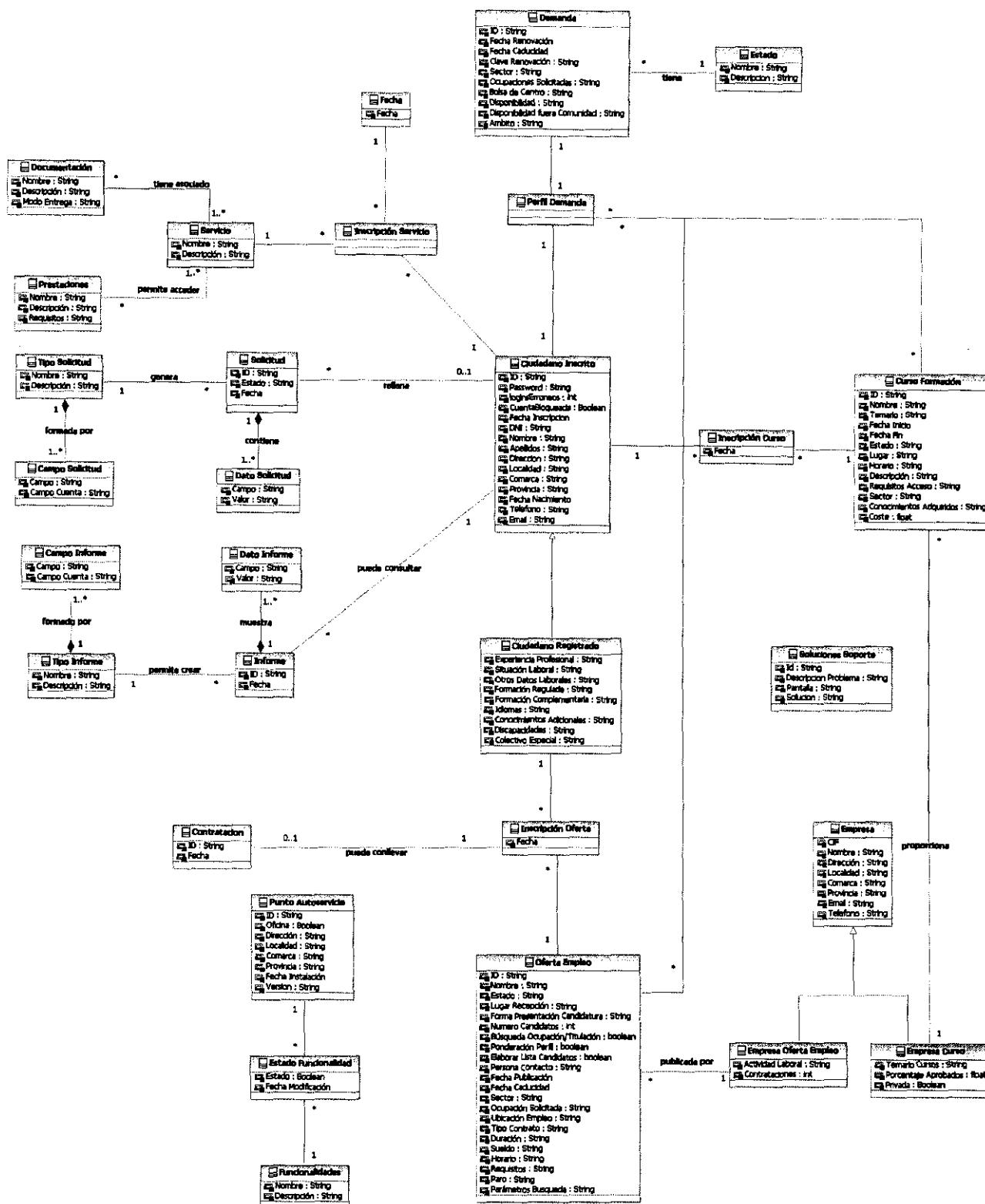


Figura 33 Diagrama de Clases

No obstante, y a pesar de que la imagen aportada muestra de una manera clara el aspecto estático del Sistema, cabe destacar una serie de decisiones que se han tomado a la hora de definir este diagrama y que facilitarán la comprensión del mismo:

- Los Informes de un Ciudadano están compuestos en su mayoría por datos que el Ciudadano aporta al crearse una cuenta en el Sistema. No obstante, por cuestiones de eficiencia y facilidad de implementación de la realización **Petición Informes** del Caso de Uso **Gestión Información**, se ha decidido crear una clase llamada **Dato Informe**, asociada a la clase **Informe**. Así pues, cada **Informe** tiene asociadas una serie de instancias de la clase **Dato Informe**, cada una de las cuales contiene un dato mostrado en el **Informe** en cuestión. Es fácilmente observable que se está guardando información duplicada, ya que la mayoría de instancias de la clase **Dato Informe** contienen valores asociados a la cuenta de un Ciudadano. Así pues, en el momento en que un Ciudadano se dé de alta en el Sistema, se deberán crear todos los datos asociados a sus Informes y cada vez que se actualicen los datos de la cuenta de un Ciudadano en la realización **Actualizar Perfil**, deberán actualizarse los datos de los Informes de dicho Ciudadano. Como se comentará más adelante, esta actualización se llevará a cabo mediante disparadores o *triggers* de Base de Datos. Cabe destacar además que, como se recoge en las restricciones de integridad del Modelo Conceptual, se establece una relación entre los datos de un **Informe** representados por la clase **Dato Informe** y los campos de dicho **Informe**, representados por la clase **Campo Informe**, de manera que el número de datos de un **Informe** no sea nunca superior al número de campos que determinan dicho **Informe**.
- Cabe destacar del mismo modo la materialización del atributo **Contrataciones** de la clase **Empresa Oferta Empleo**. Este atributo contiene el número de instancias de la clase **Contrataciones** asociadas a las **Ofertas de Empleo** publicadas por dicha empresa, es decir, el número de ciudadanos contratados por una empresa para todas las ofertas de empleo publicadas. La materialización de este atributo tiene como objetivo de facilitar la elaboración de una serie de estadísticas realizadas por el SOC.
- Por otra parte, este Diagrama de Clases pone de manifiesto las diferencias entre un **Ciudadano Inscrito** y uno **Registrado** en lo que se refiere a la información que éstos aportan al darse de alta en el SOC. Como se puede observar, las diferencias se fundamentan en que el **Ciudadano Registrado** debe aportar, además de todos los datos aportados por el **Ciudadano Inscrito**, una serie de información asociada a su formación y vida laboral.
- Cabe destacar también que es fácilmente observable que la relación que se establece entre la clase **Ciudadano Inscrito** y la clase **Demanda** es de 1-1. Rápidamente alguien podría pensar que toda la información que contiene la clase **Demanda** podría haberse incluido en la clase **Ciudadano Inscrito**. Sin embargo, se ha tomado la decisión de disgregar toda esta información en 2 clases diferentes con el objetivo de descargar de responsabilidades a clase **Ciudadano Inscrito**, clase que ya contiene de por sí una elevada carga de competencias.
- Siguiendo con las relaciones que se establecen entre las clases **Demanda** y **Ciudadano Inscrito**, cabe destacar la presencia de la clase **Perfil Demanda**. La presencia de esta clase se justifica debido a las relaciones que establece con los Cursos de Formación y las Ofertas de Empleo con el objetivo de representar aquellos Cursos y Ofertas que se ajustan al perfil del Ciudadano. Como se observará en los Diagramas de Secuencia que conforman este Modelo de Análisis, estas recomendaciones de Cursos y Ofertas que se ajustan al perfil del Ciudadano, se determinan en base al tipo de sector laboral

para el que el Ciudadano realiza su demanda de empleo y en función de los conocimientos de idiomas de dicho Ciudadano.

- Por otra parte, como se ha observado en la presencia del atributo *Contrataciones*, este Diagrama de Clases no representa únicamente la información asociada a las funcionalidades del Punto de Autoservicio, sino que contiene datos asociados a otra serie de actividades del SOC que no están presentes en los Puntos de Autoservicio pero si tienen una relación muy estrecha. Entre estas actividades cabe destacar las inscripciones que realizan los Ciudadanos en servicios ofrecidos por el SOC o la propia clase **Contrataciones**. Esta información será utilizada de manera secundaria e interna en algunas de las realizaciones de los Casos de Uso de los Puntos de Autoservicio tales como la **Eliminación de la Cuenta**, proceso en el cual se muestra al usuario un resumen de los datos que el SOC dispone de dicho usuario, incluyendo sus inscripciones en servicios.
- Del mismo modo, cabe destacar la presencia una serie de datos que están al margen de la información relacionada con los Ciudadanos y su interacción con los Puntos de Autoservicio. Estos datos representados por clases **Punto Autoservicio**, **Estado Funcionalidad** y **Funcionalidades** están asociados a las tareas de gestión, administración y monitorización de los Puntos de Autoservicio, tareas realizadas por el personal del SOC.
- Por otra parte, es fácilmente observable la presencia de una clase que no está relacionada con ninguna otra. Esta clase, llamada **Soluciones Soporte**, corresponde a la información asociada a las soluciones que los Puntos de Autoservicio ofrecen a las peticiones de soporte realizadas por los Ciudadanos. Entre los elementos que conforman esta clase cabe destacar al atributo *Pantalla*, la presencia del cual se justificará en apartados posteriores.

7.2 Realización de los Casos de Uso

Los Diagramas VOPC y los Diagramas de Secuencia de las realizaciones pretenden aportar una descripción de las interacciones que se producen internamente en el Sistema entre los diferentes componentes que conforman su arquitectura en 3 capas en cada una de las funcionalidades ofrecidas por los Puntos de Autoservicio. Cabe destacar que esta arquitectura en 3 capas, es decir, **Capa de Presentación**, **Capa de Dominio** y **Capa de Datos** se describirá con detalle en el apartado *“Tecnologías y Arquitectura del Sistema”*. No obstante, con el fin de facilitar la comprensión de este apartado, a continuación se introducirán una serie de conceptos asociados a la Arquitectura del Sistema, conceptos que serán ampliamente detallados tanto en el apartado asociado a la arquitectura y las tecnologías utilizadas comentado anteriormente, como en el apartado relativo al Modelo de Diseño o **Design Model**.

La Capa de Presentación está formada por las diferentes pantallas o **Boundaries** que componen la interficie gráfica y por un **Controlador de Presentación** encargado tanto de comunicarse con la Capa de Dominio como de crear nuevas **Boundaries**. La Capa de Dominio está definida por una serie de Controladores, uno por cada Caso de Uso, y por las diferentes clases o entidades que componen el Modelo Conceptual del Sistema. La comunicación de esta Capa de Dominio con la Capa de Datos se realizará a través del **Datamapper Hibernate** [24]. La utilización de esta **Datamapper**, así como los motivos que justifican su utilización serán explicados con detalle en los 2 apartados anteriormente comentados: *“Tecnologías y Arquitectura*

del Sistema” y “*Design Model*”. No obstante, cabe destacar que este *Datamapper* no aparece en los Diagramas de Secuencia del *Analysys Model* debido a que este Modelo tiene únicamente por objetivo mostrar las diferentes interacciones que se producen entre los componentes que forman el Sistema, sin entrar en ningún momento en detalles tecnológicos o de diseño, detalles que se reservan para los Diagramas de Secuencia que determinan el ***Design Model***.

Entrando en detalles asociados a la definición de los diagramas de ciertas realizaciones, cabe destacar que durante la creación de los Diagramas VOPC y los Diagramas de Secuencia asociados a las realizaciones **Logout** y **Petición Soporte** se han detectado ciertas peculiaridades que es necesario destacar. Estas tres realizaciones tienen en común que se pueden realizar desde diferentes lugares de la aplicación, ya que por ejemplo la **Petición de Soporte** se puede realizar desde cualquier pantalla de la aplicación. Esta característica puede inducir a pensar que resulta imposible representar gráficamente en un Diagrama las interacciones que se producen en cada una de estas realizaciones, ya que existen múltiples pantallas desde dónde se puede acceder a las funcionalidades asociadas a dichas realizaciones. Pero si observamos con detenimiento las características de estas realizaciones veremos que, si bien es cierto que pueden iniciarse desde múltiples pantallas, todas ellas se activan al seleccionar una de las opciones ubicadas en la parte superior de la pantalla, en una región que llamaremos **Barra de Herramientas**. Resulta obvio pensar que esta **Barra de Herramientas**, a pesar de las mínimas variaciones que pueda sufrir entre 2 pantallas diferentes, será única, es decir, se diseñará e implementará físicamente una única vez a pesar de que se utilice en pantallas diferentes. Así pues, podemos considerar esta **Barra de Herramientas** como un componente más de la **Capa de Presentación**, que formará parte de cada una de las pantallas del Sistema y que será el punto de inicio o **Boundary** desde la que se iniciarán las realizaciones **Logout**, y **Petición Soporte**.

Cabe destacar que en las realizaciones **Rellenar Solicitud**, **Buscar Cursos** y **Buscar Ofertas**, al igual que ha sucedido con los Diagramas Actor-Sistema, la variación de comportamiento en función de la tipología del usuario que inicia dicha realización obliga a definir, para cada una de estas 3 realizaciones, un diagrama por cada tipo de actor diferente que lleva a cabo la realización.

7.2.1 Diagramas VOPC

Los **Diagramas VOPC (View Of Participant Classes)** pretenden mostrar gráficamente y de manera concisa qué componentes del Sistema participan en una realización. Además de identificar estos componentes, los **Diagramas VOPC** muestran brevemente las relaciones que se establecen entre ellos, especificando para cada componente con que otros componentes interacciona durante la realización, sin entrar en ningún momento a detallar ni el orden ni las características de dicha interacción.

Así pues, considerando las características de la arquitectura en 3 capas anteriormente comentada, en cada **Diagrama VOPC** se identificarán:

- **Boundaries** con las que el usuario interactúa durante la realización.
- **Controladores** tanto de la Capa de Presentación como de la Capa de Dominio que participan en la realización.
- **Clases** o entidades del Modelo Conceptual que son consultadas o modificadas durante la realización.

A continuación se pueden observar una serie de **Diagramas VOPC** que se pretenden destacar.

Eliminar Cuenta

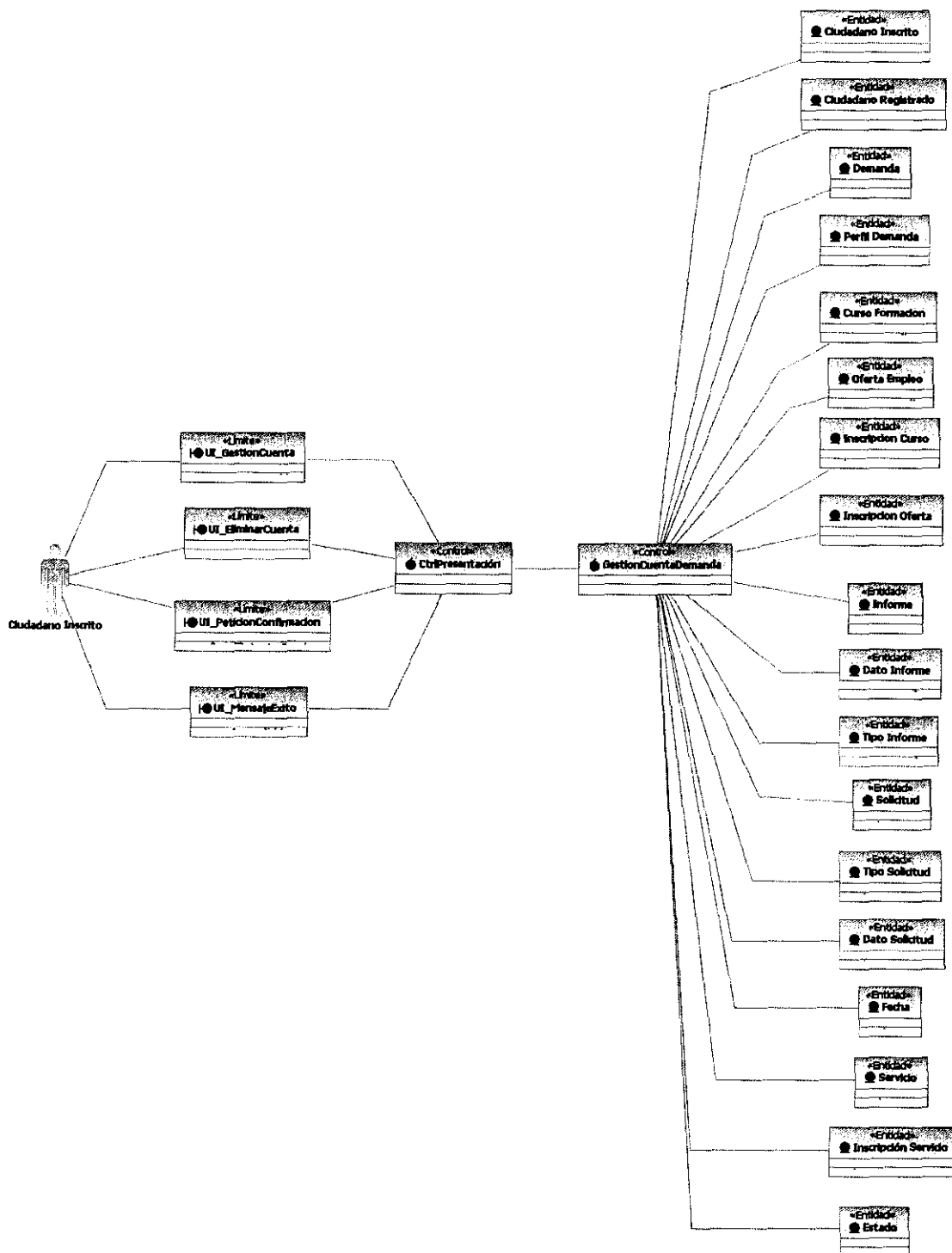


Figura 34. VOPC Eliminar Cuenta

En el **Diagrama VOPC** asociado a la realización **Eliminar Cuenta** llama poderosamente la atención la gran cantidad de Clases o Entidades del **Modelo Conceptual** que se ven afectadas por dicha realización. Esto es debido a que, como

se comenta en la descripción de la realización definida en el anexo "*Descripción de las Realizaciones de los Casos de Uso*", la eliminación de una cuenta de usuario comporta el borrado de todos los datos asociados a dicha cuenta, es decir, inscripciones, informes, solicitudes...

Por otra parte, como se ha comentado previamente, el diagrama define brevemente las interacciones que se producen entre los diferentes componentes que participan en la realización, mostrando de una manera clara y concisa que el orden en que se producen estas interacciones es el siguiente:

Boundary-Controlador Presentación-Controlador Dominio-Clases de Dominio

Buscar Ofertas

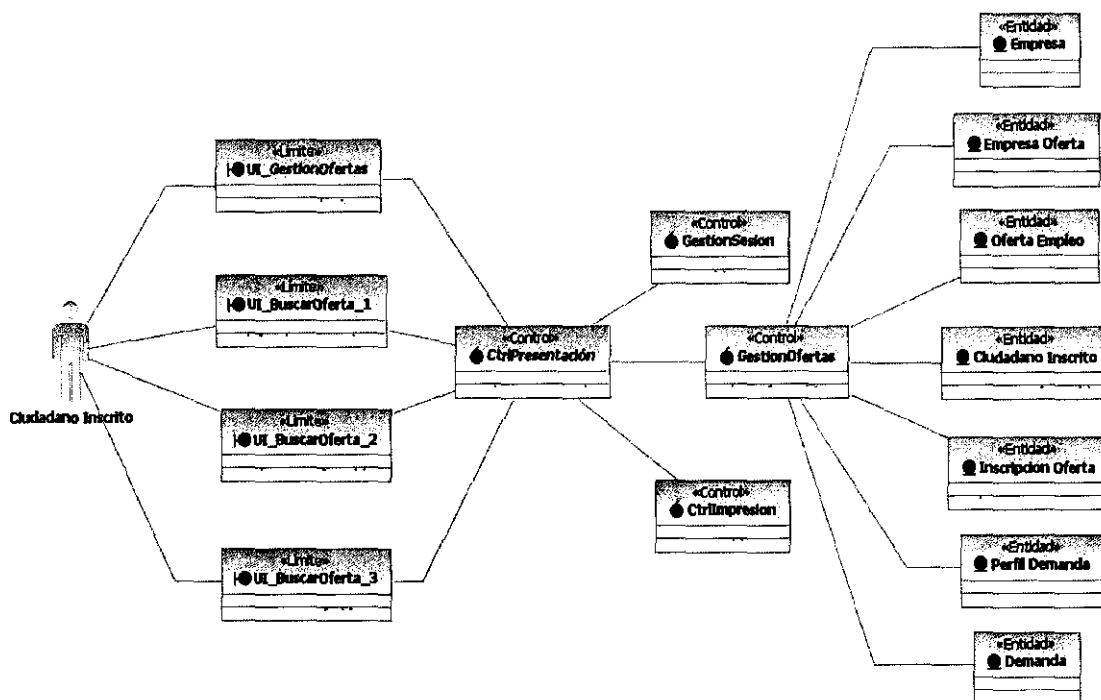


Figura 35. VOPC Buscar Ofertas

Se ha querido destacar también el **Diagrama VOPC** asociado a la realización **Buscar Ofertas del Ciudadano Inscrito**. En esta realización, como ya se ha observado tanto en la descripción textual de la misma como en el Diagrama Actor-Sistema, se define una **relación de inclusión** con el Caso de Uso **Gestión Sesión** y una de **extensión** con el Caso **Impresión Datos**. Cada una de estas relaciones comporta la existencia de un **Controlador de Dominio** adicional en el Diagrama VOPC, motivando que el diagrama de esta realización esté formado por 3 **Controladores de Dominio**, la presencia de dos de los cuales, *GestionSesion* y *CtrlImpresion*, se fundamenta en dichas relaciones.

Logout

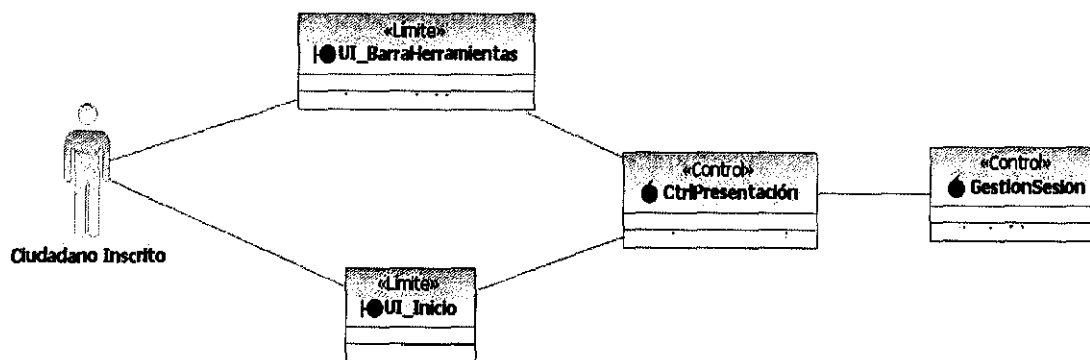


Figura 36. VOPC Logout

Por último, se ha querido destacar el **Diagrama VOPC** de la realización **Logout** por 2 motivos fundamentales. En primer lugar, por la presencia de la **Boundary Barra de Herramientas** comentada anteriormente y en segundo lugar porque se trata de una realización en la que no se accede ni modifica ninguna Entidad del dominio, particularidad que comporta la ausencia de dichas Entidades en el **Diagrama VOPC**.

En caso de que se deseen examinar los **Diagramas VOPC** del resto de realizaciones que conforman los Casos de Uso, consultar en el anexo "**Software Architecture Document**" el subapartado *Casos de Uso y Realizaciones* del apartado *Analysis Model*.

7.2.2 Diagramas de Secuencia

Dado que uno de los objetivos fundamentales del *Analysis Model* es realizar una descripción lógica de la implementación encargada de cumplir los requisitos funcionales del Sistema, los **Diagramas VOPC** resultan insuficientes para alcanzar este objetivo. Para ello, se definirá un **Diagrama de Secuencia** para cada una de las realizaciones del Sistema destinado a mostrar de manera clara y detallada las interacciones que se producen entre los diferentes componentes y entidades del Sistema. Esta descripción detallada tratará de obviar ciertos detalles tecnológicos o de diseño, como por ejemplo la interacción con el *Datamapper Hibernate*; reservando estos detalles para los diagramas que determinan el **Design Model** que se definirá posteriormente.

Dada la envergadura y las dimensiones de estos diagramas, resulta del todo imposible adjuntar la mayoría de ellos en esta documentación. Así pues, se elegirá un diagrama cuyas dimensiones permitan que sea mostrado en esta documentación para observar de manera global las diferentes interacciones que se producen entre los componentes del Sistema durante una realización. Por otra parte, se seleccionarán fragmentos de otros diagramas que contengan aspectos que se deseen destacar o comentar. En caso de que se quieran observar el resto de **Diagramas de Secuencia** definidos, se ha adjuntado a esta documentación un CD que contiene los diferentes modelos definidos mediante la herramienta *software Rational Software Modeller*.

A continuación se muestra el **Diagrama de Secuencia** asociado a la realización **Acceso Información**.

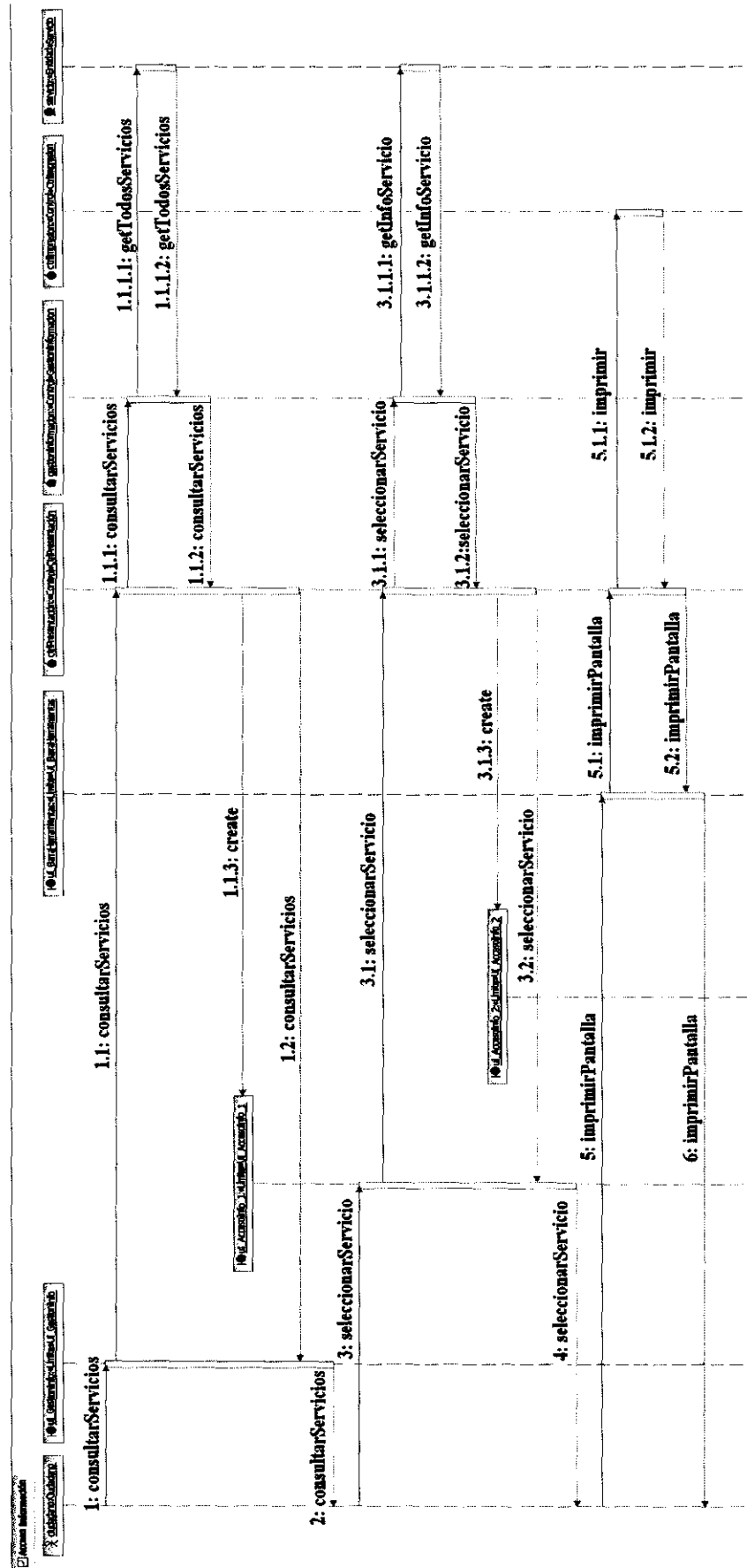


Figura 37. Analysis Model – Diagrama Secuencia Acceso Información

A continuación se muestran una serie de fragmentos de los diagramas de otras realizaciones. Cada fragmento está asociado a un concepto o característica que se desea destacar:

- **Creación de Pantallas:** Como se ha comentado previamente, según la arquitectura definida, el **Controlador de Presentación** es el encargado de crear las distintas pantallas con las que interacciona el usuario a lo largo de una realización.

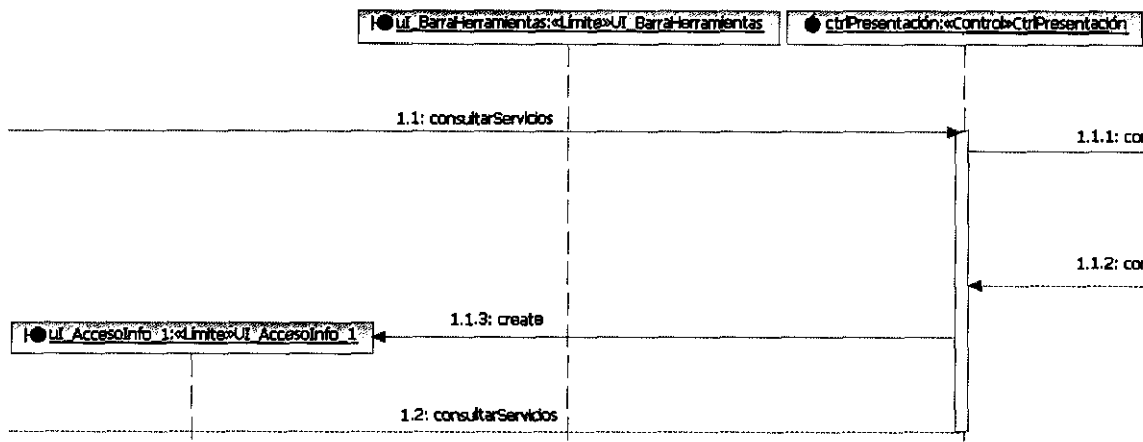


Figura 38. Diagrama Secuencia Análisis – Crear Pantallas

- **Relaciones de Inclusión y Extensión:** Las diferentes relaciones de inclusión y extensión definidas en los Casos de Uso del Sistema se muestran de una manera explícita y esclarecedora en los Diagramas de Secuencia del *Analysis Model*. Estas relaciones implican la reutilización de un diseño asociado a otra funcionalidad, diseño que debe ser tratado como si de una caja negra se tratase, es decir, se debe invocar al código asociado a esta funcionalidad sin entrar a valorar los detalles de dicho código. Así pues, estas relaciones de inclusión y extensión se representarán en un Diagrama de Secuencia como una llamada a la funcionalidad que determina dicha relación, llamada que se realizará al **Controlador de Dominio** adecuado, es decir, aquel controlador que conoce o puede acceder a dicha funcionalidad. A continuación se muestra la relación de inclusión definida en la realización **Buscar Ofertas** con el Caso de Uso **Gestión Sesión** con el objetivo de ilustrar esta descripción:

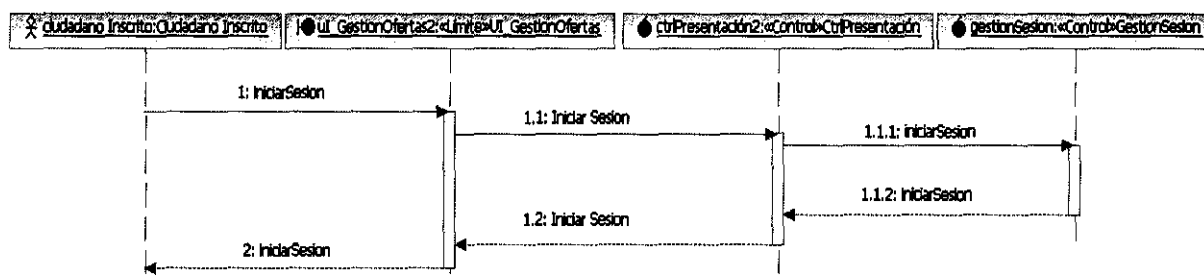


Figura 39. Diagramas Secuencia Análisis – Relaciones de Inclusión

Como se puede observar en la imagen, simplemente el **Controlador de Presentación** solicita al **Controlador de Dominio** pertinente, en este caso **GestiónSesion**, la funcionalidad requerida. Los detalles de cómo el **Controlador de Dominio** ejecuta las acciones pertinentes para llevar a cabo esta funcionalidad es totalmente indiferente desde el punto de vista de la realización **Buscar Ofertas**.

- **Actualizar Perfil implica actualizar Informes:** Como se ha comentado previamente, debido a la materialización de los datos de los que conforman los Informes, la modificación del perfil de un Ciudadano comporta la actualización de los datos de sus Informes. Como se llevará a cabo esta actualización es algo que se decidirá en el **Design Model**.

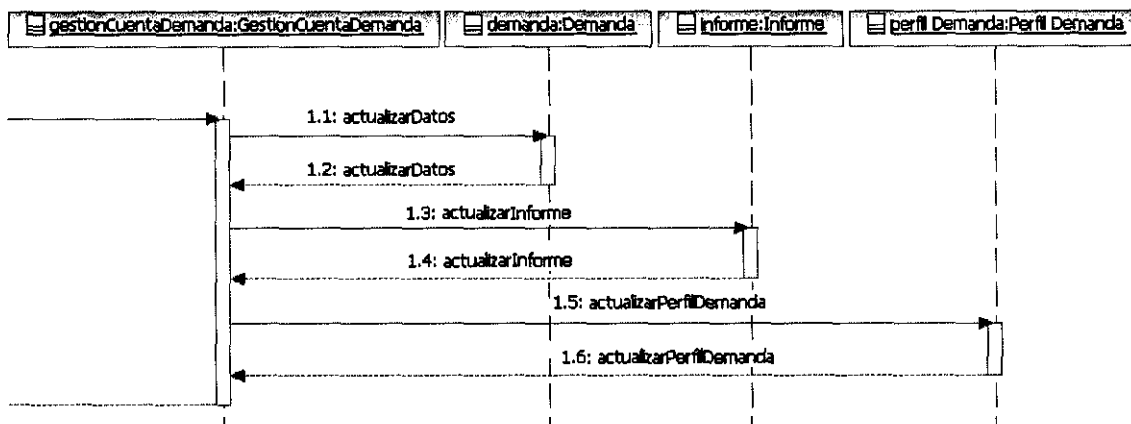


Figura 40. Diagrama Secuencia Análisis – Actualizar Informes

- **Renovar Demanda y el Informe DARDO:** Cabe destacar que existe otra realización del Sistema que comporta la actualización de los Informes de un Ciudadano. Como se ha comentado en la descripción textual de las realizaciones, la **Renovación de la Demanda** comporta actualizar el **Informe DARDO**.

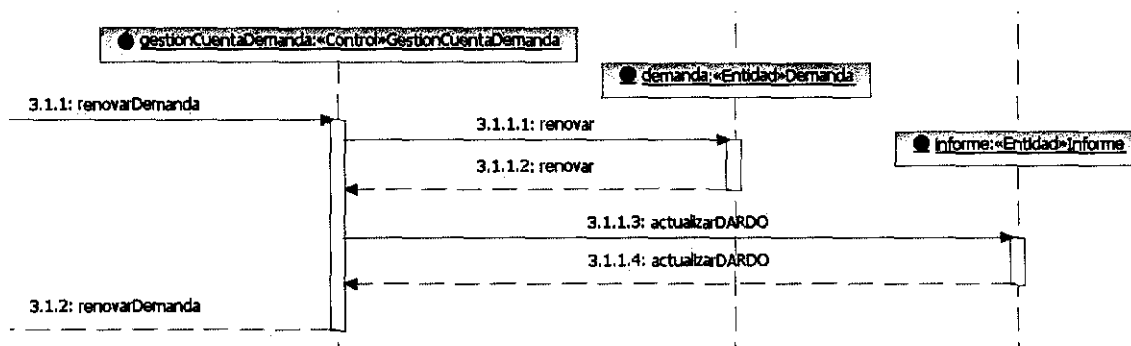


Figura 41. Diagrama Secuencia Análisis – Informe DARDO

- **Wizards Formularios:** Como se ha comentado en diversas ocasiones a lo largo de esta documentación, gran parte de los formularios de los Puntos de Autoservicio se han dividido en varias páginas dando lugar a una estructura conocida como *wizard*. Para gestionar la información introducida en un *wizard*, se ha optado por ir **acumulando** esta información aportada en cada página del formulario de manera que, hasta que no se procesa la información de la última página del *wizard*, no se realizan las modificaciones pertinentes en las clases de dominio. A continuación se muestra un fragmento de la realización **Inscripción Abreviada** donde se puede ver perfectamente esta acumulación de información y como, tras procesar la última página del formulario, se empiezan a realizar las diferentes creaciones de nuevas instancias de las entidades que conforman la Capa de Negocio del Sistema.

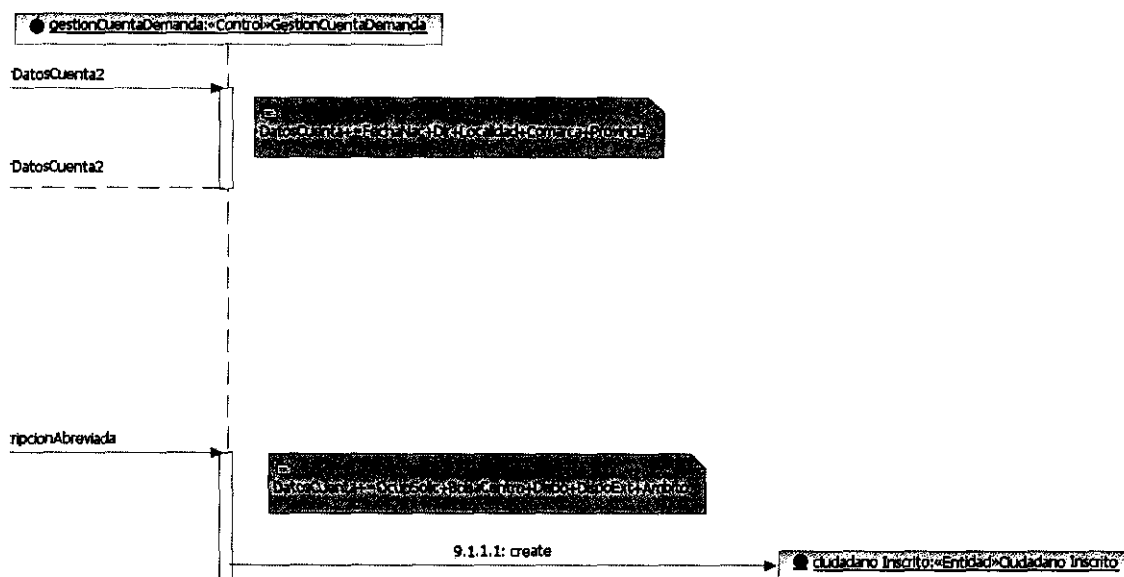


Figura 42. Diagrama de Secuencia Análisis - Wizard

- **Crear Cuenta implica crear diversas instancias de clases diferentes:** Como se puede observar en el Diagrama de Clases mostrado anteriormente, la cuenta de un Ciudadano está asociada a una gran cantidad de clases. Así pues, en el momento de crear una nueva cuenta, es decir, una nueva instancia de la clase Ciudadano Inscrito, se deberán crear y asociar a este Ciudadano Inscrito diversas instancias de otras clases tales como Demanda, Perfil Demanda, Informe o Ciudadano Registrado. En la siguiente imagen, se muestra un fragmento del Diagrama de Secuencia asociado a la realización **Inscripción Abreviada** donde se refleja la creación de estas instancias.

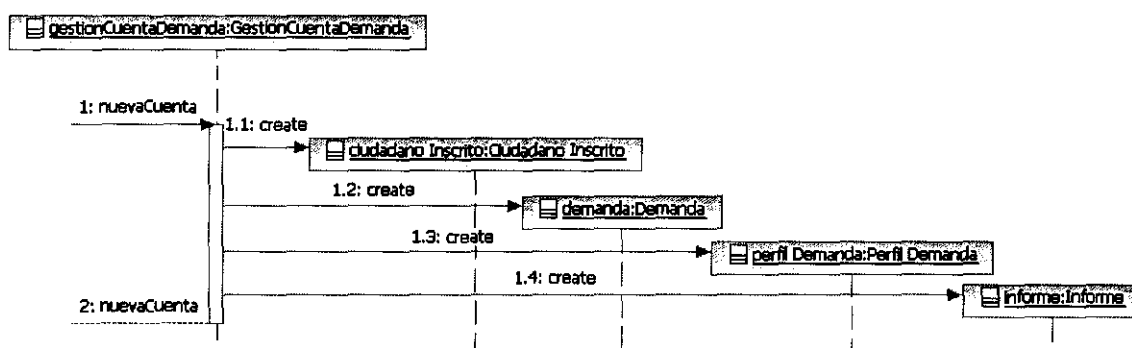


Figura 43. Diagrama Secuencia Análisis – Crear Cuenta

- **Eliminar Cuenta implica eliminar diversas instancias de clases diferentes:** Del mismo modo que la creación de una cuenta implica crear diversas clases, la eliminación comporta el borrado de todas las clases asociadas a dicha cuenta. En el **Diagrama VOPC** de la realización **Eliminar Cuenta** mostrado anteriormente pueden observarse todas las clases de dominio que se ven afectadas por esta eliminación. En caso de que se deseen observar los detalles de este **Diagrama de Secuencia**, consultar el Modelo de Análisis definido mediante *Rational Software Modeller* que ha sido adjuntado a esta documentación.
- **Barra de Herramientas:** La definición del componente **Barra de Herramientas** resulta imprescindible para la especificación de los Diagramas de Secuencia asociados a las realizaciones **Logout** y **Petición de Soporte**. A pesar de que

esta **Barra de Herramientas** es un componente que forma parte de todas las pantallas del Sistema, con el objetivo de facilitar la comprensión de los diagramas, dicha barra es utilizada como si de una *Boundary* más se tratase. A continuación se muestra un pequeño ejemplo de utilización de esta **Barra de Herramientas**.

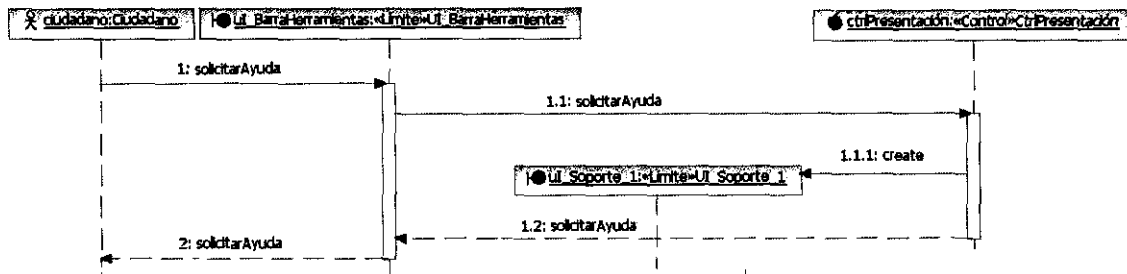


Figura 44. Diagrama Secuencia Análisis – Barra de Herramientas

8 Design Model

El **Design Model** pretende completar las tareas iniciadas durante el Modelo de Análisis describiendo pormenorizadamente cada una de las realizaciones que conforman los Casos de Uso, definiendo, por tanto, los detalles tecnológicos y de diseño del Sistema. Así pues, partiendo de este Modelo de Análisis y del análisis de requisitos, se pretende obtener una **abstracción del código fuente** del Sistema.

A pesar de que el componente principal del **Design Model** son los Diagramas de Secuencia de las realizaciones, se han realizado una serie de tareas previas a la definición de estos diagramas con el objetivo de facilitar y optimizar el diseño de éstos. Entre esta serie de tareas cabe destacar la normalización del **Modelo Conceptual**, con el objetivo de obtener un diseño que pueda ser implementado mediante una tecnología relacional de Bases de Datos, y la evaluación o estudio de los **Patrones de Diseño** que son susceptibles de ser aplicados en las realizaciones del Sistema. La aplicación de estos patrones debe permitir obtener un diseño del Sistema basado en la **flexibilidad, modularidad y reutilización**.

A partir de esta evaluación de los Patrones de Diseño, se ha procedido a definir los **Diagramas de Secuencia** del Modelo de Diseño. Estos diagramas permitirán obtener una descripción detallada de cada una de las realizaciones del Sistema, considerando la arquitectura y los detalles tecnológicos que determinan el mismo. Estos detalles tecnológicos comprenden tanto la arquitectura en 3 capas que define el Sistema como la utilización del **Datamapper Hibernate**. Las características asociadas a la inclusión de este **Datamapper** en los Diagramas de Secuencia se describen con detalle en el subapartado de "**Persistencia**". Cabe destacar, que como se ha mencionado en el Modelo de Análisis, todos los detalles relativos a la Arquitectura del Sistema serán ampliamente detallados en el apartado "**Tecnologías y Arquitectura del Sistema**".

8.1 Patrones de Diseño

La Ingeniería del Software ofrece unas prácticas y metodologías para el diseño de sistemas distribuidos con una persistencia basada en una Base de Datos. Una vez se dispone de un esquema modular y estructurado del Sistema, existen una serie de Patrones de Diseño creados para facilitar ciertas funcionalidades o potenciar ciertas características del Sistema mediante algunas estructuras definidas. A continuación se describen los patrones aplicados para el Sistema de Puntos de Autoservicio, así como una serie de decisiones que se han tomado asociadas a dicha aplicación.

8.1.1 Patrón Iterador

El Patrón Iterador esta asociado a una serie de realizaciones del **Design Model** tales como:

- **Inscripción Abreviada** del Caso de Uso Gestión Cuenta y Demanda
- **Crear Cuenta** del Caso de Uso Gestión Cuenta y Demanda.
- **Buscar Cursos** del Caso de Uso Gestión Formación.
- **Buscar Ofertas** del Caso de Uso Gestión Ofertas.

- **Activar Servicio** del Caso de Uso Gestión Punto Autoservicio.
- **Desactivar Servicio** del Caso de Uso Gestión Punto Autoservicio.

Estas realizaciones efectúan un **recorrido** por cada uno de los elementos de una **colección de objetos**. El Patrón Iterador nos permite recorrer esta colección actuando en cada iteración sobre uno de los elementos de esta colección.

A continuación, podemos observar la utilización del Patrón Iterador en la realización Activar Servicio. En esta realización, se deben recorrer todas las instancias de la Clase Estado Funcionalidad para obtener el atributo Estado y el Punto de Autoservicio asociado a dicho Estado.

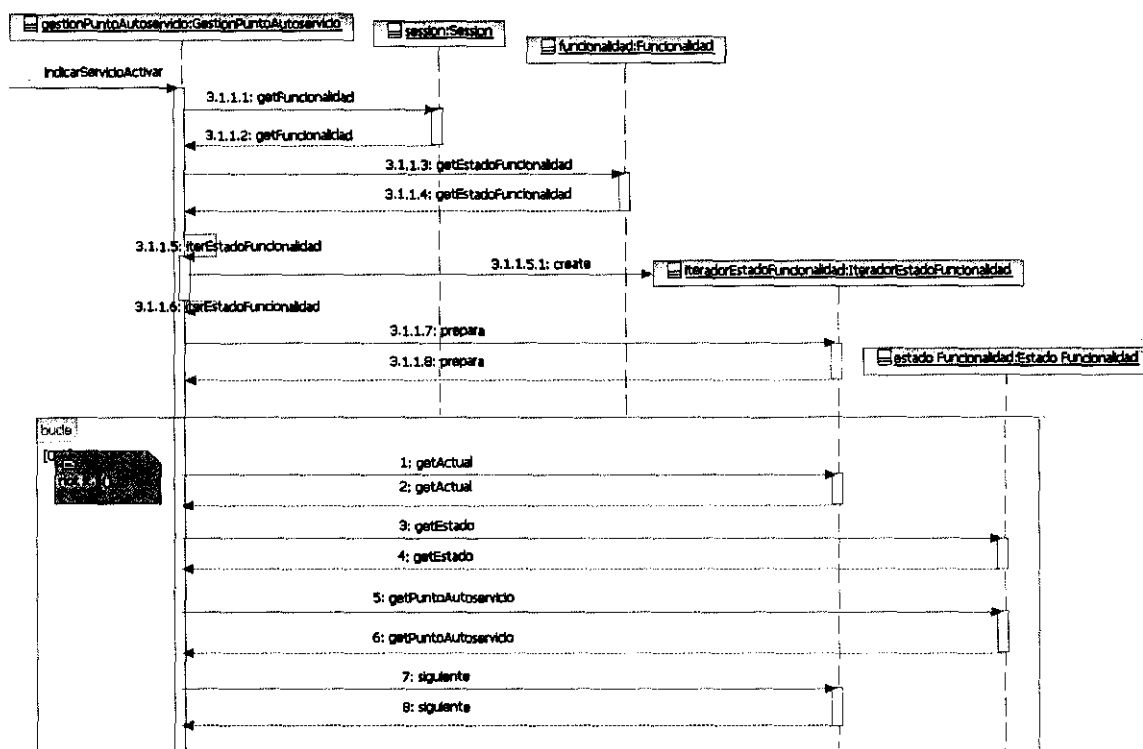


Figura 45. Design Model – Patrón Iterador

El uso del patrón implica la creación de la clase Iterador, la cual ejerce de interficie para el acceso secuencial.

8.1.2 Patrón Controlador

Nos encontramos ante un sistema orientado a la interacción externa a través de los terminales de Autoservicio y que consta de una estructura organizada mediante capas. Estas características nos llevan a plantearnos que elemento u objeto será el encargado de capturar los eventos realizados por los usuarios en las diferentes pantallas del Sistema, así como procesar dichos eventos llevando a cabo las acciones pertinentes.

La respuesta a estas dudas la encontramos en el **Patrón Controlador Caso de Uso**. Dicho controlador implica la existencia de un **objeto diferente para cada uno** de los **Casos de Uso** del Sistema, el cual se encargará de gestionar toda la interacción asociada a cada una de las realizaciones del Caso de Uso.

En la Capa de Presentación encontramos el **Controlador de Presentación**, que **capturará los eventos externos** y se los comunicará a su homólogo correspondiente de la Capa de Dominio, el Controlador del Caso de Uso en cuestión.

Así pues, por un lado tenemos el Controlador de Presentación, cuya responsabilidad es **crear y gestionar las diferentes pantallas** del Sistema, así como capturar los eventos que en dichas pantallas se producen. Por otro lado, tenemos un conjunto de controladores asociados a cada uno de los Casos de Uso encargados de **ejecutar las acciones pertinentes** como respuesta a dichos eventos.

A continuación podemos observar un ejemplo de esta serie interacciones en la realización Petición Soporte del Caso de Uso Soporte y Atención al Cliente:

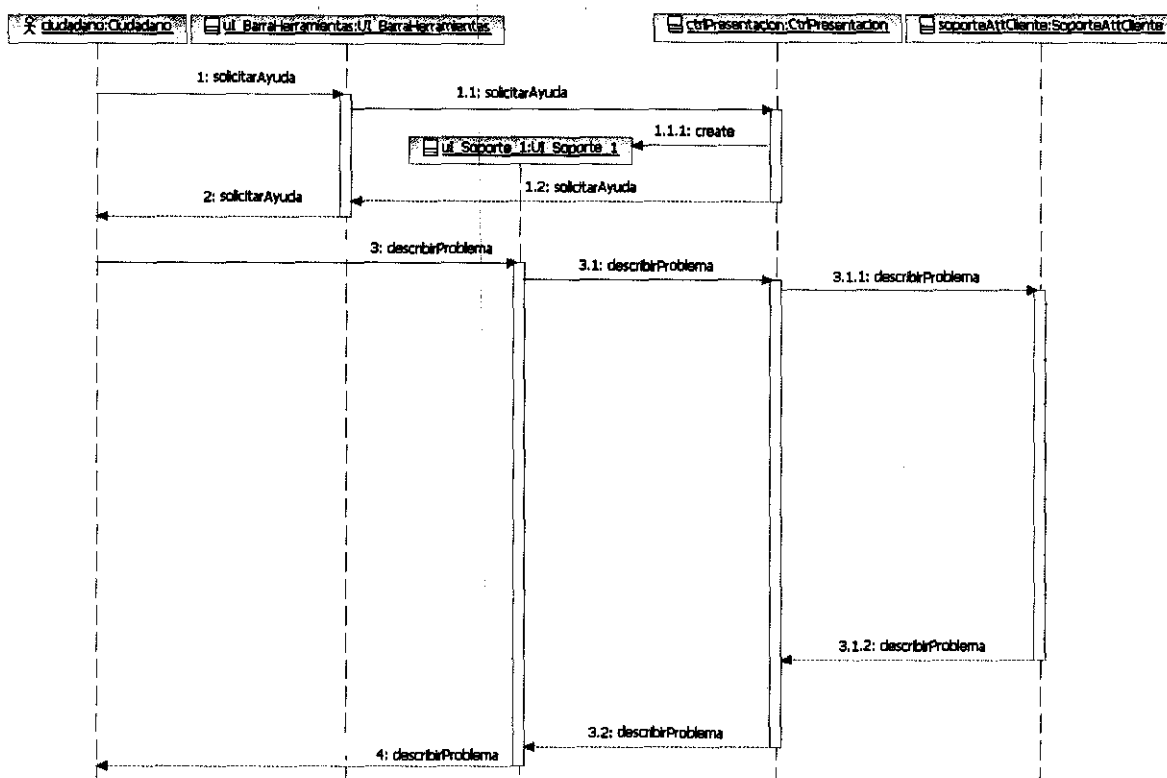


Figura 46. Design Model – Patrón Controlador Caso de Uso

8.1.3 Patrón Experto y Creador

Estos patrones hacen referencia a la **asignación de responsabilidades** de las operaciones y la **creación de instancias** respectivamente. La correcta asignación de dichas responsabilidades mantiene el encapsulamiento y el comportamiento distribuido entre las clases del sistema, lo que permite **reducir** el posible **acoplamiento** y **aumentar su cohesión**.

La aplicación del Patrón Controlador ya facilita y orienta hacia la correcta aplicación del patrón, ya que define los controladores que tendrán la responsabilidad de distribuir las operaciones y de realizar las creaciones pertinentes.

8.1.4 Otros patrones de diseño

Cabe destacar que se ha estudiado pormenorizadamente cada una de las realizaciones del **Design Model** evaluando la necesidad o no de la aplicación de otros patrones de diseño. Al final se ha desechado esta posibilidad debido a que las características de estas realizaciones no requerían la aplicación del patrón o bien se ha optado por un mecanismo alternativo. A continuación se citan una serie de casos relevantes en los que se ha descartado la aplicación de cierto patrón de diseño y se aporta una justificación para desechar dicha aplicación:

1. **Inscripción Oferta - Patrón Observador:** Cuando un Ciudadano se inscribe en una Oferta de Empleo, automáticamente se envía una **notificación por email** a la empresa que publica dicha Oferta. Se ha descartado al aplicación del Patrón Observador debido a que tenemos **un único observador**, la empresa que recibe la notificación y la eficacia del Patrón Observador radica en el hecho que el sujeto que soporta la actualización debe notificar esta actualización a más de un observador llegando incluso a desconocer el número de éstos.
2. **Inscripción Curso – Patrón Observador:** Al igual que ocurre en la realización Inscripción Oferta, cuando un Ciudadano se inscribe en un Curso de Formación se envía una notificación a la empresa que ofrece dicho Curso. Se ha descartado la aplicación del Patrón Observador por la misma razón que en la realización Inscripción Oferta.
3. **Actualizar Perfil – Patrón Observador:** Como se ha comentado en diversas ocasiones a lo largo de esta documentación, la actualización de los datos de la cuenta de un Ciudadano comporta la **actualización de los datos** de sus Informes. Este hecho se presta para la aplicación del Patrón Observador, actuando como Observadores los datos de los Informes. Pero finalmente se ha decidido realizar esta actualización de los datos de los Informes mediante una serie de **triggers** de Base de Datos que se detallarán en posteriores apartados.
4. **Eliminar Cuenta – Patrón Observador:** En la realización Eliminar Cuenta nos encontramos con una situación similar a la Actualización de Perfil. La eliminación de una cuenta de un Ciudadano comporta la **eliminación de toda la información** asociada a esta cuenta. De nuevo, estamos ante una situación que se presta a la aplicación del Patrón Observador, pero al igual que en la Actualización del Perfil, se ha optado por resolver esta eliminación de datos asociados a la cuenta mediante una serie de **triggers** de Base de Datos.

8.2 Persistencia

Una parte importante de los diagramas que conforman el **Design Model** es la comunicación de la Capa de Negocio del Sistema con la Capa de Persistencia. Como se ha comentado previamente en esta documentación, se ha seleccionado el **Datamapper Hibernate** para gestionar esta comunicación. Los detalles que fundamentan dicha decisión se describen con detalle en el apartado “**Tecnologías y Arquitectura del Sistema**”

La utilización de esta tecnología se puede observar en los diagramas que conforman el **Design Model**, tanto en los **VOPC's** en los que aparece la Capa de Persistencia, como en los **Diagramas de Secuencia**, en los que se introducen las clases de acceso propias de **Hibernate SessionFactory** y **Session**.

La clase **SessionFactory** es la encargada crear instancias de la clase **Session**, la cual dispone de una serie de funciones que permiten acceder y manipular los datos almacenados persistentemente en el Sistema. De entre estas funciones destinadas a gestionar los datos almacenados, destacamos las siguientes:

- **get:** Función que permite realizar consultas de datos almacenados en la Base de Datos del Sistema.
- **flush:** Función que permite registrar persistentemente los cambios realizados.
- **close:** Función que permite cerrar la comunicación con **Hibernate**.

En las siguientes imágenes se puede observar la interacción con las clases **SessionFactory** y **Session**, así como la utilización de estas 3 funciones:

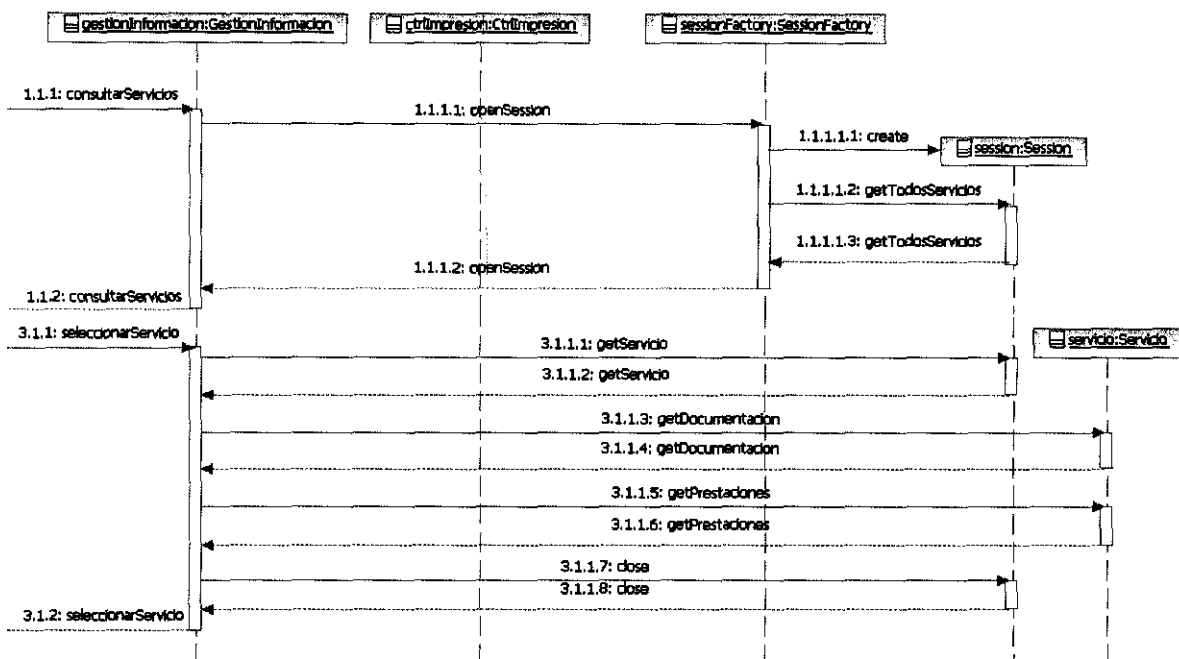


Figura 47. Design Model – Funciones get y close de Hibernate

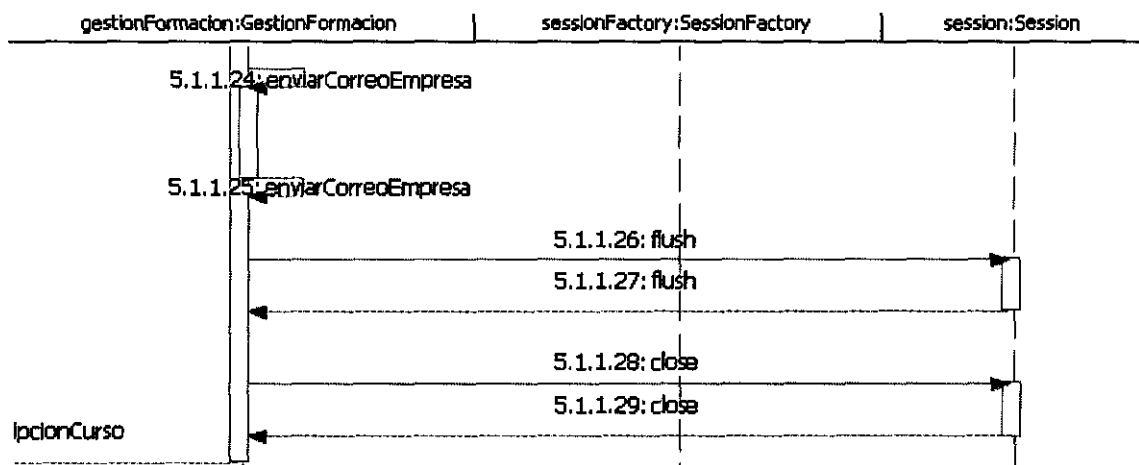


Figura 48. Design Model – Funciones flush y close de Hibernate

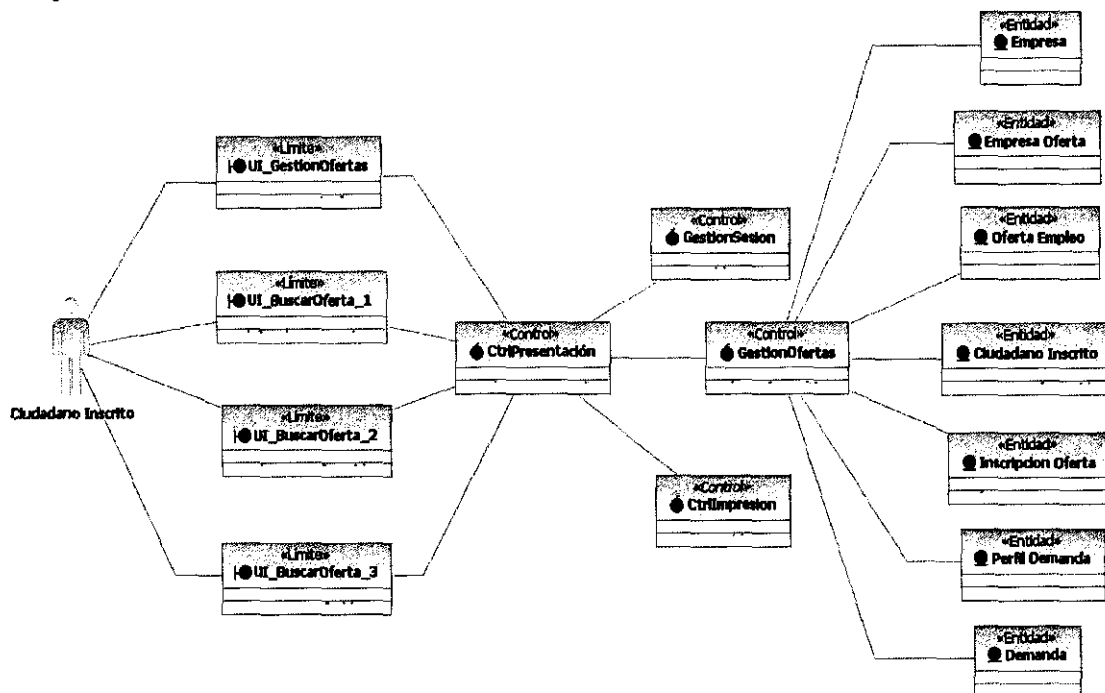
8.3 Diagramas VOPC

Cabe destacar que en el **Design Model** se produce una modificación de los **Diagramas VOPC** definidos en el **Analysis Model**, actualizando dichos diagramas en función de las decisiones de diseño tomadas y los detalles tecnológicos asociados a la arquitectura en el Sistema.

Así pues, para el Sistema de Puntos de Autoservicio, cabe destacar que la aplicación del **Datamapper Hibernate** conlleva la abstracción o transparencia de la interacción que se produce entre el **Controlador de Dominio** y las Clases o Entidades que conforman dicho Dominio. Esta abstracción aparece representada en estos **Diagramas VOPC** sustituyendo cada una de las Clases de Dominio que aparecen en los **Diagramas VOPC** del **Analysis Model** por un nuevo controlador que llamaremos **Controlador de Persistencia** y que representa simplemente la función desarrollada por el **Datamapper** anteriormente mencionado.

A continuación se muestran los **Diagrama VOPC**, tanto del **Analysis Model** como del **Design Model**, correspondientes a la realización **Buscar Oferta** con el objetivo de observar las diferencias anteriormente comentadas.

Analysis Model



Design Model

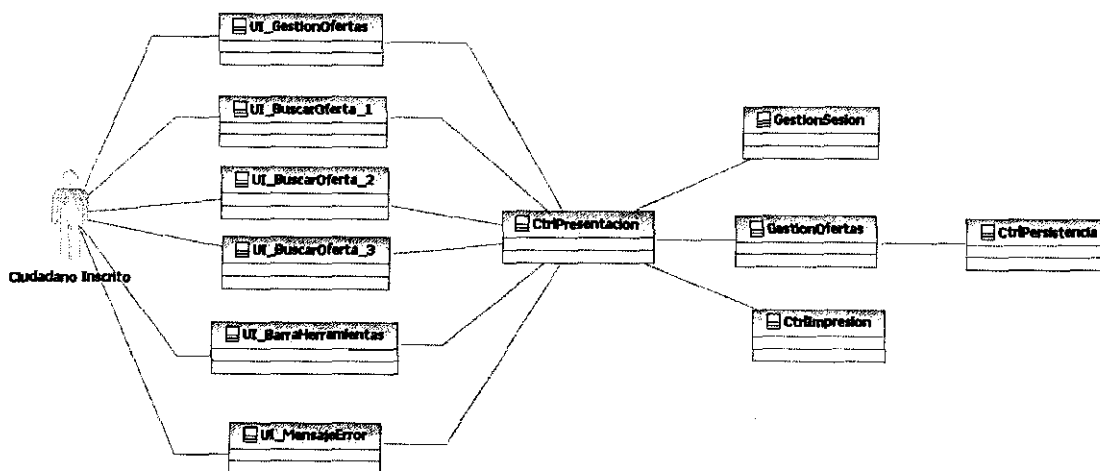


Figura 49. Design Model VOPC – Gestión Ofertas

En caso de que se deseen consultar los Diagramas VOPC del resto de realizaciones, se han incluido en el anexo “*Software Architecture Document*”, en el apartado correspondiente al *Design Model*.

8.4 Diagramas de Secuencia

Los Diagramas de Secuencia constituyen el elemento fundamental del **Design Model**, describiendo de una manera detallada las interacciones que se producen entre los diferentes componentes de la arquitectura del Sistema en cada una de las realizaciones que conforman los Casos de Uso del mismo. A diferencia de los Diagramas de Secuencia del **Analysis Model**, en los diagramas del **Design Model** se describen estas interacciones incluyendo cada uno de los detalles tecnológicos que definen la arquitectura del Sistema, tales como la utilización de **Hibernate**, y las decisiones de diseño que se han tomado.

Así pues, se trata de unos diagramas que describen las interacciones de una manera más precisa, no limitándose únicamente a mostrar la secuencia de interacciones que se producen entre los diferentes componentes, sino describiendo de manera detallada y minuciosa cada uno de los procesos internos que se realizan en el Sistema para culminar con éxito cada una de las realizaciones. El nivel de detalle de estos diagramas permite que puedan ser considerados como una **abstracción del código fuente** del Sistema.

Al igual que ha sucedido con el Modelo de Análisis, las dimensiones de los Diagramas de Secuencia del **Design Model** impiden que puedan ser adjuntados a esta documentación. No obstante, se escogerá un diagrama cuyas dimensiones posibiliten su inclusión en esta documentación y se seleccionarán fragmentos de diagramas cuyo contenido se pretenda destacar. En caso de que se deseen consultar el resto de diagramas del **Design Model**, pueden examinarse con detalle utilizando el CD adjuntado a esta documentación. Así pues, a continuación se muestra el **Diagrama de Secuencia** de la realización **Acceso Información**. Se ha seleccionado este diagrama debido a que es el mismo que se ha incluido en el apartado del **Analysis Model**, con lo cual facilita la observación de las diferencias que presentan ambos modelos entre las que cabe destacar:



- La comunicación que se establece entre el Controlador de Dominio y **Hibernate** con el objetivo de gestionar la comunicación entre la Capa de Dominio y la Capa de Persistencia.
- La indicación de fragmentos iterativos u opcionales mediante la estructura correspondiente.
- La concreción de cada uno de los detalles de las interacciones. Esto es fácilmente observable en la realización **Acceso Información**, ya que en dicha realización se informa al usuario de las características de cada uno de los servicios ofrecidos por el SOC. En el diagrama del *Analysis Model* la obtención de esta información se limitaba a una petición a la clase *Servicio* llamada *getInfoServicio*. En el diagrama del *Design Model* se especifica exactamente que información que se quiere obtener de la clase servicio, es decir, cada uno de los datos de este servicio, es decir, la documentación requerida y las prestaciones ofrecidas.

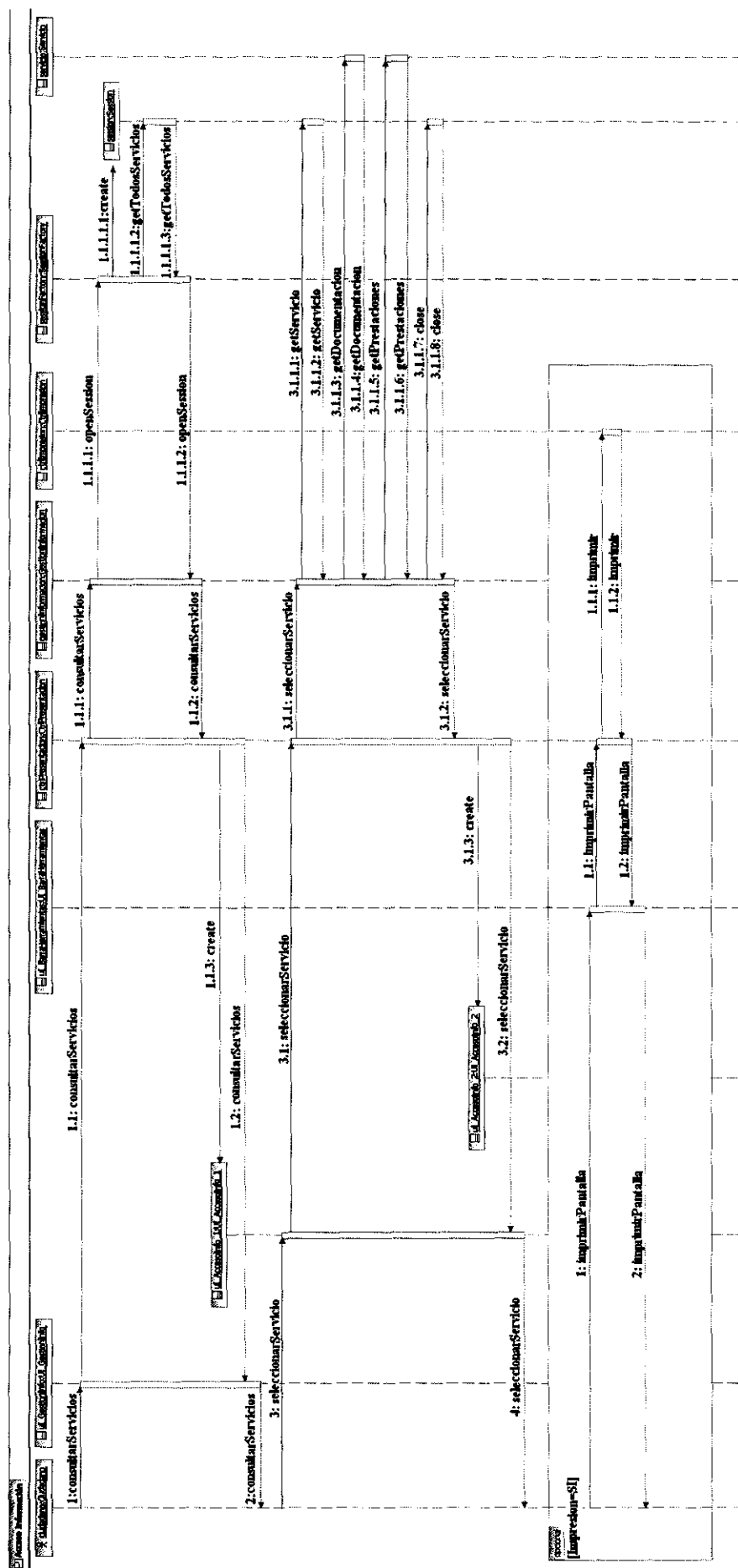


Figura 50. Design Model – Diagrama Secuencia Acceso Información

Uno de los elementos de los Diagramas de Secuencia del **Design Model** que no aparece en la realización **Acceso Información** y que merece ser destacado es la notificación de errores al usuario, notificación que no se realizaba en los diagramas del **Analysis Model**. Esta notificación de errores se produce mediante una **Boundary** llamada **MensajeError**, **Boundary** que en la mayoría de los casos aparece dentro de un fragmento de diagrama opcional, debido a que los casos de error suelen estar asociados a flujos alternativos de la realización. A continuación se muestra el fragmento de un diagrama del **Design Model** en el que aparece esta **Boundary**.

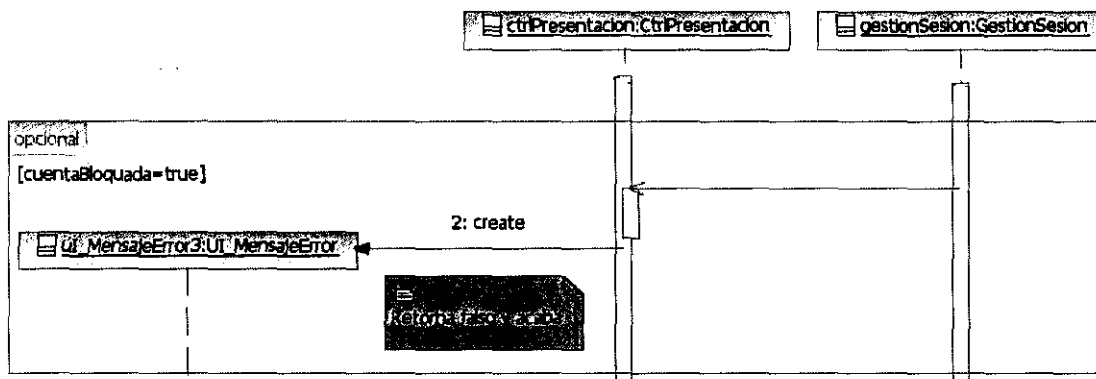


Figura 51. Design Model – Boundary Mensaje Error

Por otra parte, ya se ha remarcado en diversas ocasiones la importancia del **Datamapper Hibernate** en los Diagramas de Secuencia del **Design Model**. No obstante, cabe destacar de nuevo el relevante papel que juega la funcionalidad **get** de dicho **Datamapper** en el diseño de cada una de las realizaciones. Esta funcionalidad está presente en prácticamente la totalidad de ellas, facilitando enormemente el diseño de las mismas, dado que permite realizar consultas de una variada complejidad. A continuación se muestran una serie de fragmentos de los Diagramas en donde se muestran consultas de diferente complejidad.

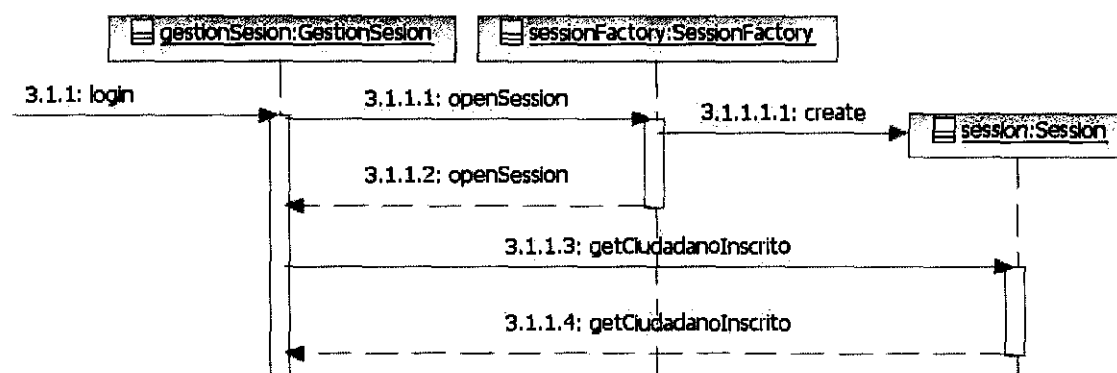


Figura 52. Design Model – Clases Session Factory y Session

En la imagen anterior podemos observar la simple petición de una instancia concreta de la clase **Ciudadano Inscrito** utilizando el identificador de dicho ciudadano.

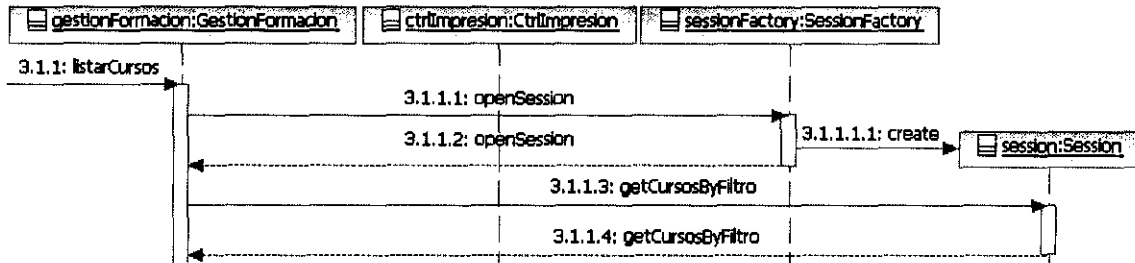


Figura 53. Design Model. Funcion get Hibernate

En cambio, en esta otra imagen puede observarse la utilización de la función **get** para realizar una consulta más compleja, concretamente para determinar el conjunto de Cursos de Formación que cumplen una serie de restricciones especificadas en un filtro de búsqueda.

Cambiando de tercio, cabe destacar una serie de particularidades asociadas a la inscripción o registro de ciudadanos en el Sistema mediante las realizaciones **Crear Cuenta** e **Inscripción Abreviada**. Como se ha comentado previamente, se puede afirmar que los datos aportados en la **Inscripción Abreviada** no son más que un **subconjunto** de los que se deben aportar a la hora de **Crear** una **Cuenta**. Así pues, resulta obvio pensar que las pantallas que forman la realización de la inscripción serán también un subconjunto de las pantallas que definen la creación una cuenta. Concretamente, existen 2 pantallas correspondientes a las páginas 3 y 4 del formulario de creación de una cuenta que no forman parte de la **Inscripción Abreviada**. Así pues, la única diferencia apreciable en los **Diagramas de Secuencia** de ambas realizaciones es que, en la creación de una cuenta, tras procesar los datos asociados a la página número 2 del formulario, el Sistema muestra la página número 3 de dicho formulario, mientras que al realizar un proceso de inscripción, muestra directamente la página 5 del mismo. Esta diferencia entre ambas realizaciones puede observarse en las siguientes imágenes.

Inscripción Abreviada

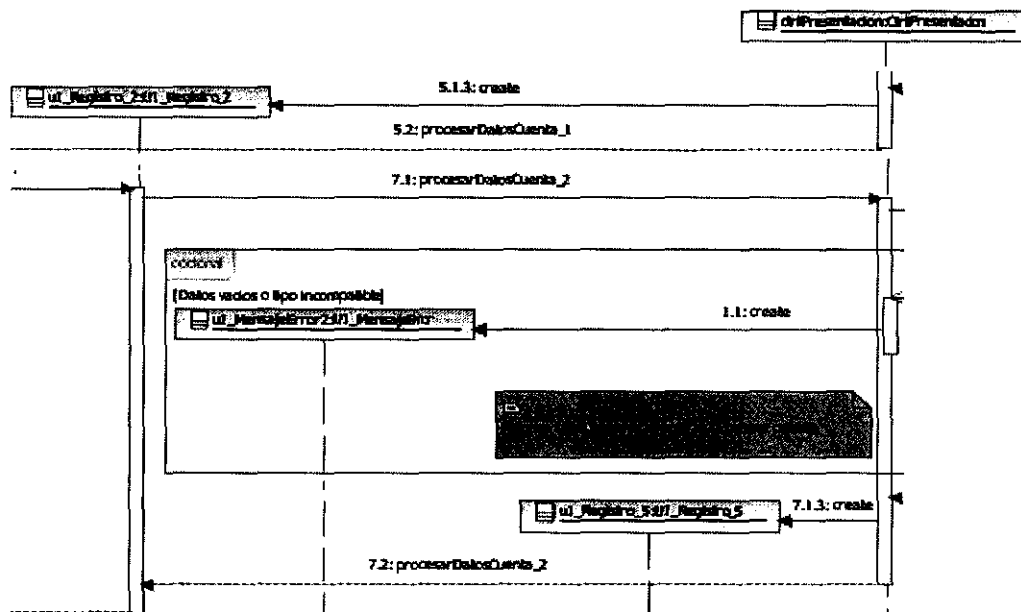


Figura 54. Design Model -- Inscripción Abreviada

Crear Cuenta

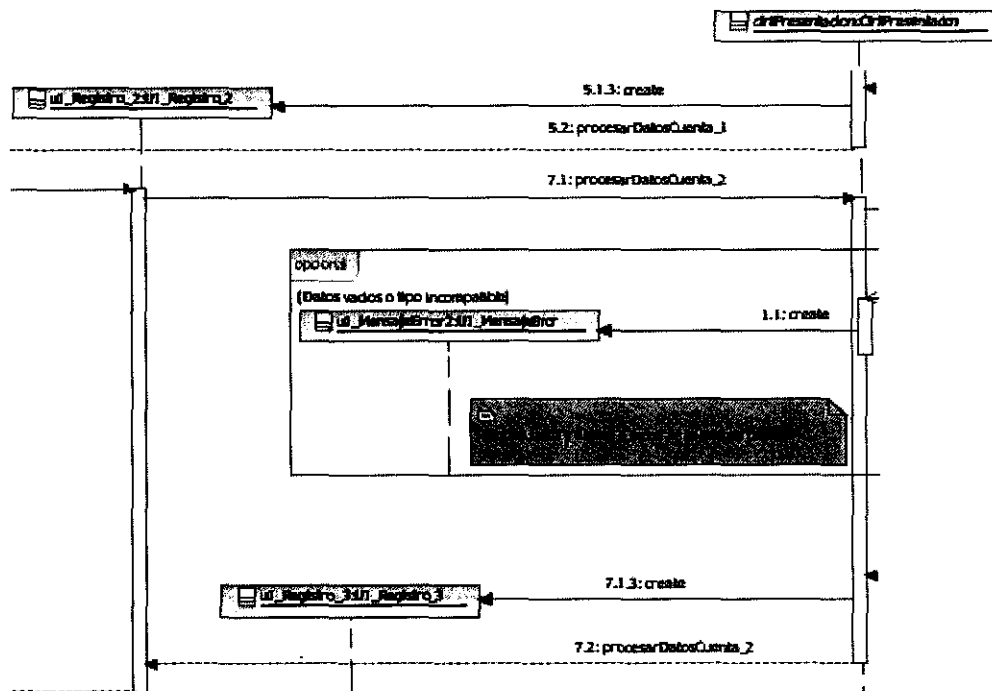


Figura 55. Design Model – Crear Cuenta

Otro de los aspectos destacados asociado a los procesos de **Inscripción Abreviada y Crear Cuenta** es la definición de los **Informes**. Las razones que justifican la creación de **Informes** en estas realizaciones han sido comentadas en diversas ocasiones a lo largo de esta documentación. Para llevar a cabo este proceso de creación de informes, cabe destacar el relevante papel que juega un atributo de la Clase **CampoInforme**, concretamente el atributo **CampoCuenta**. Este atributo indica, para un campo de un informe, si el valor de dicho campo está asociado o no a un dato aportado por el ciudadano en el proceso de registro en el Sistema y, en caso de que lo esté, en que clase o entidad del Sistema se encuentra dicho valor. Así pues, el proceso que se sigue para crear estos informes es tan simple como:

- Para cada **Tipo de Informe**:
 - Se crea el **Informe**.
 - Se **asocia al Ciudadano** que está dándose de alta en el Sistema.
 - Para cada **Campo del Informe**:
 - Se crear una instancia de tipo **DatoInforme** y se asocia al **Informe** creado.
 - El valor de este **DatoInforme** depende de si el campo en cuestión esta asociado o no a un valor introducido por el Ciudadano en el proceso de registro.
 - Si esta asociado, se consulta dicho valor a través del atributo **CampoCuenta** descrito anteriormente.
 - Si no está asociado, se crea el **DatoInforme** con el valor por defecto estipulado.

Dadas las dimensiones de los Diagramas de Secuencia asociados a las realizaciones **Inscripción Abreviada y Crear Cuenta**, ha resultado del todo imposible poder incluir ni tan siquiera una imagen del fragmento asociado a la creación de

informes. No obstante, en caso de que se deseen consultar estos diagramas, ambos están incluidos en el CD adjuntado a esta documentación.

Otra realización en la que juegan un papel importante los **Informes** de un Ciudadano es en la **Actualización del Perfil**. Como se ha comentado en diversas ocasiones, al actualizar el perfil de un Ciudadano, se deben actualizar los datos de los informes de éste. Pero en el apartado asociado a la aplicación de **Patrones de Diseño** se ha justificado que dicha actualización se realice mediante una serie de **triggers** de Base de Datos. Así pues, en el **Diagrama de Secuencia** de la realización **Actualizar Perfil** no se realiza en ningún momento una modificación de los datos de los Informes, limitándose únicamente a actualizar los datos del perfil del Ciudadano. Por tanto, esta actualización de los datos de los informes mediante **triggers** simplifica enormemente el Diagrama de Secuencia de la realización **Actualizar Perfil**.

Existe otra realización en la que los **triggers** de Bases de Datos desempeñan una función destacada. Se trata lógicamente de la realización **Eliminar Cuenta**. La eliminación de la cuenta de un Ciudadano comporta la supresión de todos los datos asociados a ella. Como se ha comentado en el apartado asociado a los **Patrones de Diseño**, dicha supresión se realizará mediante una serie de **triggers** de Base de Datos. La definición de estos **triggers** facilita y simplifica enormemente el **Diagrama de Secuencia** asociado a la realización **Eliminar Cuenta**, ya que éste debe limitarse a realizar únicamente la eliminación de la instancia de la clase **Ciudadano Inscrito**. En la siguiente imagen, se muestra el fragmento del **Diagrama de Secuencia** en el que se realiza esta eliminación, la única que se lleva a cabo en todo el diagrama.

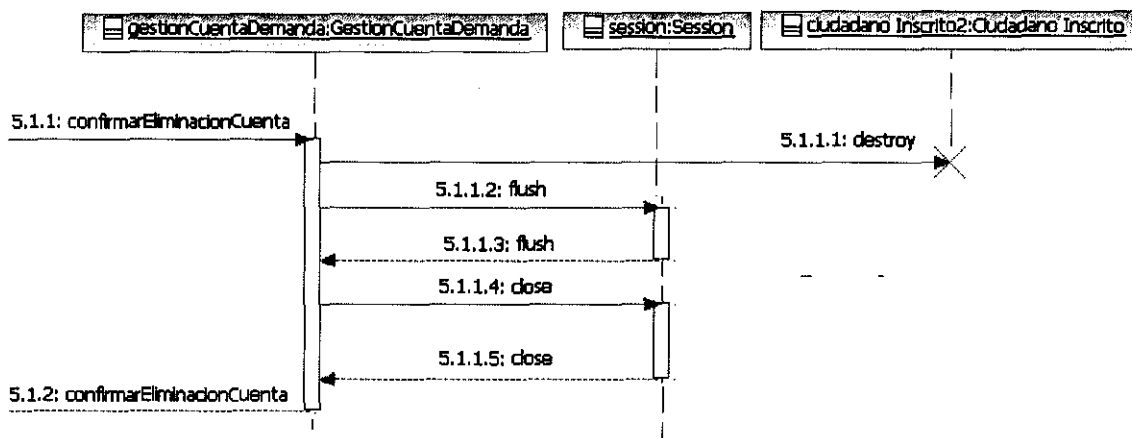


Figura 56. Design Model – Eliminar Cuenta

Otras de las clases de Dominio asociadas a un Ciudadano que debe ser destacada es la clase **PerfilDemanda**. Esta clase permite representar aquellas **Ofertas de Empleo y Cursos de Formación** que se ajustan al perfil de un Ciudadano. Así pues, esta clase **PerfilDemanda** permitirá recomendar a un Ciudadano Cursos y Ofertas de Empleo cuando éste realice una búsqueda. Lógicamente, la definición de esta clase **PerfilDemanda** deberá realizarse cuando el Ciudadano se dé de alta en el Sistema, es decir, en las realizaciones **Inscripción Abreviada y Crear Cuenta**. El criterio que se ha seguido para determinar que cursos y ofertas se ajustan al perfil del Ciudadano se basa fundamentalmente en 2 criterios:

- Sector laboral para el que el Ciudadano realiza su demanda de Empleo.
- Idiomas que conoce el Ciudadano.

Cuando el Ciudadano se da de alta en el Sistema especifica para qué sector laboral realiza preferentemente la demanda de empleo. Además, a cada Curso de Formación y Oferta de Empleo se determina el sector laboral al que están asociados. Por otra parte, los usuarios de tipo **Ciudadano Registrado**, al **Crear una Cuenta** en el Sistema, indican los idiomas que conocen y hablan con fluidez. Así pues, a los **Ciudadanos Inscritos** se les recomendarán aquellas Ofertas de Empleo y Cursos de Formación cuyo sector laboral corresponde al sector para el que estos Ciudadanos realizan su demanda de empleo. Además, a los **Ciudadanos Registrados** se les recomendarán **Cursos de Formación de Idiomas** si el número de idiomas que conocen es reducido, concretamente inferior a 3. A continuación, se muestra el **Diagrama de Secuencia** de la realización **Inscripción Abreviada** en el cual se determinan los **Cursos de Formación** que se ajustan al perfil del Ciudadano. En casos de que se deseen examinar el resto de procesos asociados a las recomendaciones de Cursos de Formación y Ofertas de Empleo, consultar el CD adjuntado a esta documentación donde podrán observarse detenidamente cada uno de los procesos detallados anteriormente.

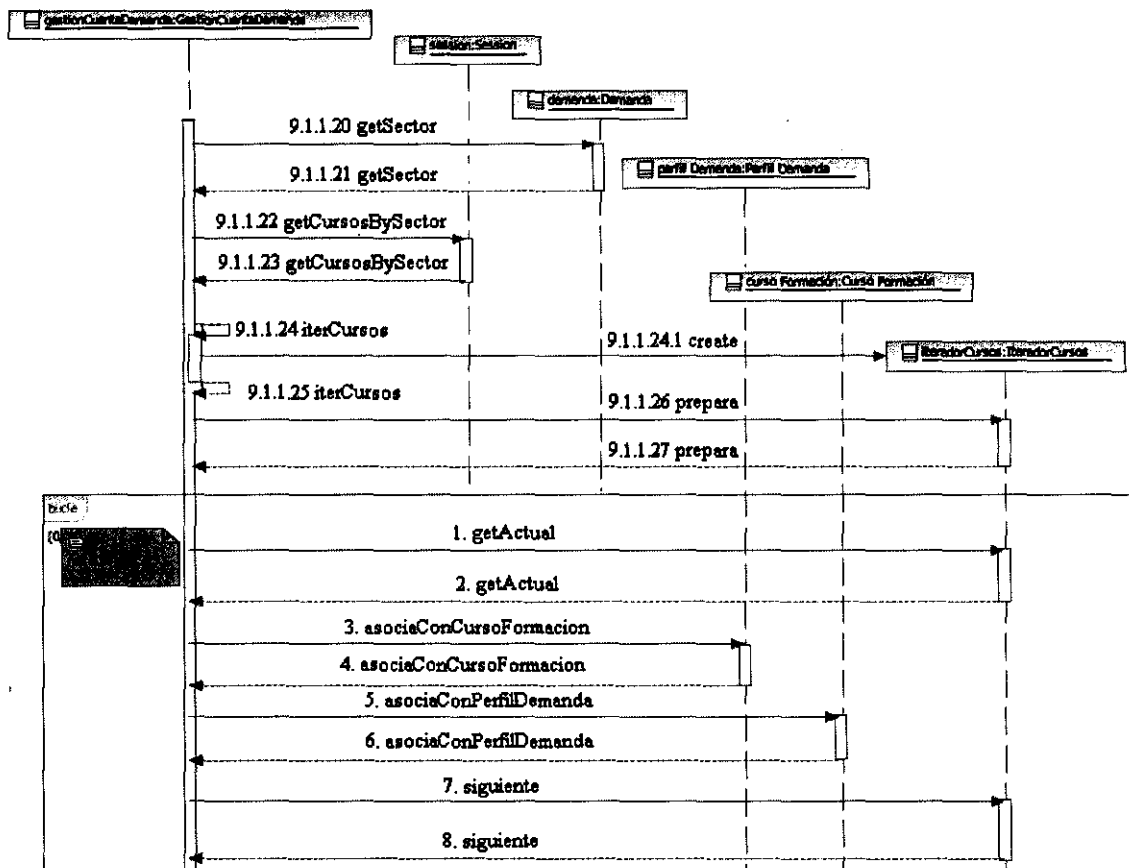


Figura 57. Design Model – Inscripción Abreviada Cursos Recomendados

Cambiando totalmente de tercio, como se ha remarcado a lo largo de esta documentación, el Sistema de Puntos de Autoservicio gestiona una gran cantidad de datos personales de los usuarios. La gestión de estos datos requiere establecer una serie de medidas de seguridad. En el **Design Model** pueden observarse algunas de estas medidas, entre las que cabe destacar el **bloqueo de una cuenta** de usuario al intentar iniciar una sesión de manera errónea 3 veces consecutivas. Es decir, si un usuario intenta iniciar una sesión en el Sistema introduciendo 3 veces consecutivas un *password* incorrecto, el Sistema bloqueará la cuenta del usuario impidiendo el acceso

a la misma. El bloqueo de la cuenta, así como el número de intentos erróneos de inicio de sesión que se han realizado de manera consecutiva, quedan registrados en la clase de dominio **CiudadanoInscrito**, concretamente en los atributos **CuentaBloqueada** y **LoginsErroneos**. Para desbloquear la cuenta, el usuario deberá ponerse en contacto con el SOC. A continuación, se muestra el fragmento del **Diagrama de Secuencia** de la realización **Login** en el que se realiza el bloqueo de una cuenta.

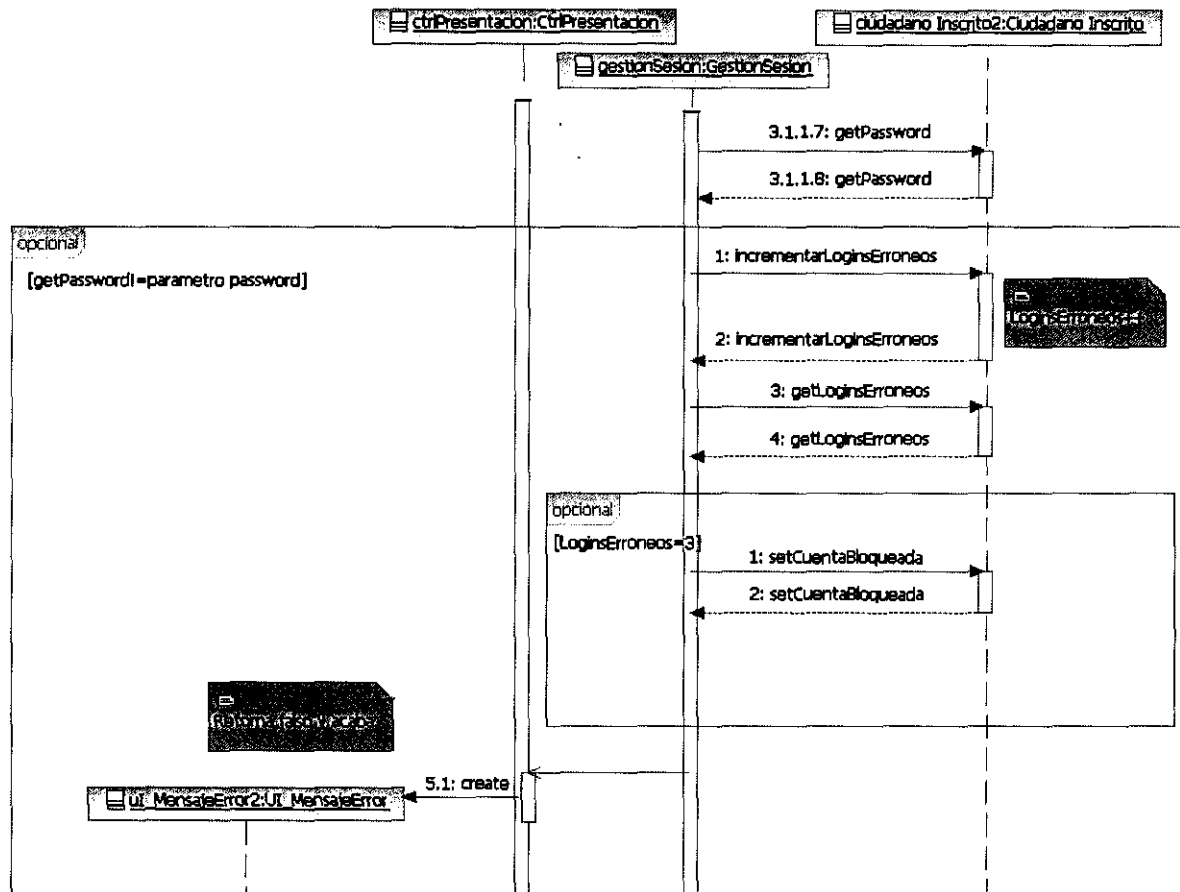


Figura 58. Design Model – Bloqueo de una cuenta

Otra de las funcionalidades de los Puntos de Autoservicio cuyo diseño debe ser destacado es la **Petición de Soporte**. Como bien es sabido, los usuarios de los Terminales de Autoservicio no están acostumbrados a interactuar con sistemas automatizados. Así pues, es altamente probable que les surjan dudas derivadas de su utilización del Sistema. Estas dudas deben ser resueltas por el sistema de ayuda de los terminales y, por tanto, cabe prestar una especial atención a esta funcionalidad de soporte. La funcionalidad **Petición de Soporte** tiene un funcionamiento muy simple:

- El usuario solicita ayuda al Sistema.
- El usuario describe su problema.
- El Sistema proporciona una solución al problema.

En el **Modelo Conceptual** del Sistema se puede observar fácilmente que existe una clase de dominio que no está asociada a ninguna otra. Se trata de la clase **SolucionesSoporte**. Esta entidad contiene simplemente una serie de soluciones asociadas a la descripción de un problema.

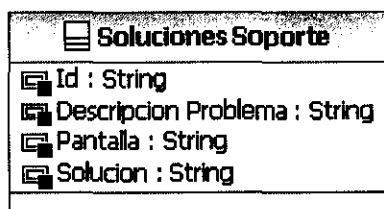


Figura 59. Design Model – Clase Soluciones Soporte

Entre los atributos que conforman esta clase destaca especialmente el atributo **Pantalla**. Se ha considerado que es muy probable que las descripciones de los problemas o dudas de los ciudadanos no sea lo suficientemente precisa como para encontrar la solución correcta a su problema. Así pues, este atributo **Pantalla** debe facilitar la búsqueda de esta solución indicando desde que pantalla del terminal se ha producido la Petición de Soporte, es decir, este **identificador de pantalla** acota de alguna manera los posibles problemas que pueda tener el usuario y aporta información adicional sobre dicho problema, facilitando de esta manera la obtención de la solución adecuada. A continuación se muestra el fragmento del Diagrama de Secuencia de la realización **Petición Soporte** en el que juega un papel destacado la clase **Session** de **Hibernate**, encargada de realizar la búsqueda de soluciones.

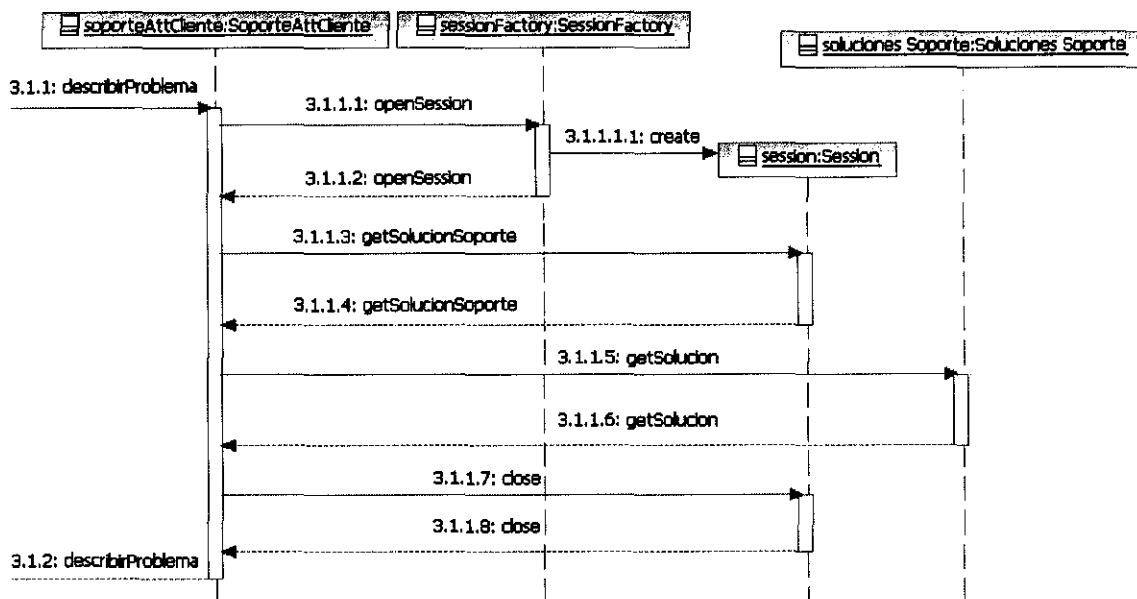


Figura 60. Design Model – Busqueda soluciones soporte

Por otra parte, en la realización **Rellenar Solicitud** para Ciudadanos Inscritos, el Sistema debe completar automáticamente una serie de campos de la Solicitud cuyo valor ya es conocido. La obtención de estos valores presenta grandes similitudes con el proceso de creación de Informes de un Ciudadano mediante el atributo **CampoCuenta**. Como se puede observar en el Diagrama de Clases del Sistema, cada una de las solicitudes del Sistema está formada por una serie de campos, **CampoSolicitud**. Esta clase **CampoSolicitud** contiene un atributo llamado **CampoCuenta**, que al igual que sucede con el proceso de creación de Informes, indica si el valor ese campo de la solicitud está asociado a alguno de los datos del ciudadano ya conocido. Así pues, cuando el Sistema muestra a un Ciudadano Inscrito una Solicitud, para cada campo de esta Solicitud, consulta el atributo **CampoCuenta** y, en caso de que este atributo le indique que el valor de ese campo ya es conocido, consulta dicho el valor en la clase de dominio pertinente. En el CD adjuntado a esta

documentación se puede consultar el **Diagrama de Secuencia** de la realización **Rellenar Solicitud**, observando detalladamente el proceso que se ha descrito.

Finalmente, como se puede observar en la descripción textual de las realizaciones de los Casos de Uso, cuando un Ciudadano se inscribe en un Curso de Formación o en una Oferta de Empleo, se envía una notificación por correo electrónico a la empresa que imparte dicho Curso u ofrece dicha Oferta. Este correo contiene simplemente una serie de datos del Ciudadano que realiza la inscripción tales como nombre y apellidos y una serie de datos de contacto. A continuación se muestra un fragmento del Diagrama de Secuencia de la realización **Inscripción Oferta** en el que se puede observar el envío de esta notificación:

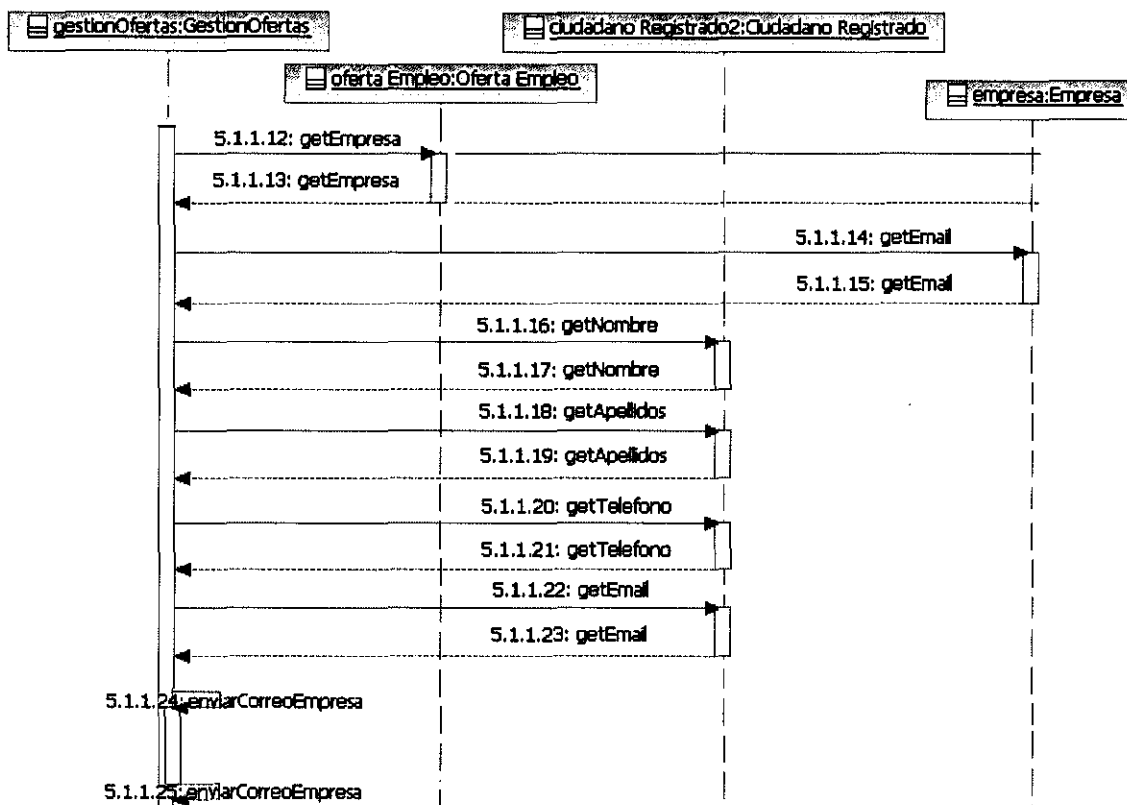


Figura 61. Design Model – Notificar Inscripciones a empresas

9 Tecnologías y Arquitectura del Sistema

Además de realizar el análisis y diseño de cada una de las funcionalidades ofrecidas en los Puntos de Autoservicio, este proyecto pretende establecer el marco tecnológico sobre el que se desarrollarán estas funcionalidades. Este sistema de Puntos de Autoservicio se sustentará en la clásica **arquitectura en 3 capas: Capa de Presentación, Capa de Dominio y Capa de Persistencia**. A lo largo de este apartado se describirán las tecnologías utilizadas en cada una de las capas, así como las herramientas y métodos que permiten la comunicación entre estas capas.

A continuación se muestra una descripción de cada uno de los principales componentes que conforman la arquitectura del Sistema.

9.1 Descripción del Sistema

En la imagen que se muestra a continuación, podemos ver un diagrama que resume globalmente y de manera muy resumida la arquitectura del Sistema. El objetivo de este diagrama es simplemente definir la base de esta arquitectura estableciendo el punto de partida sobre el que se desarrollarán las definiciones y explicaciones que conforman este apartado. Cabe destacar que cada uno de los componentes que aparecen en la imagen será descrito con detalle a lo largo de este apartado. Además se especificarán las tecnologías que permiten establecer una comunicación entre cada uno de estos componentes.

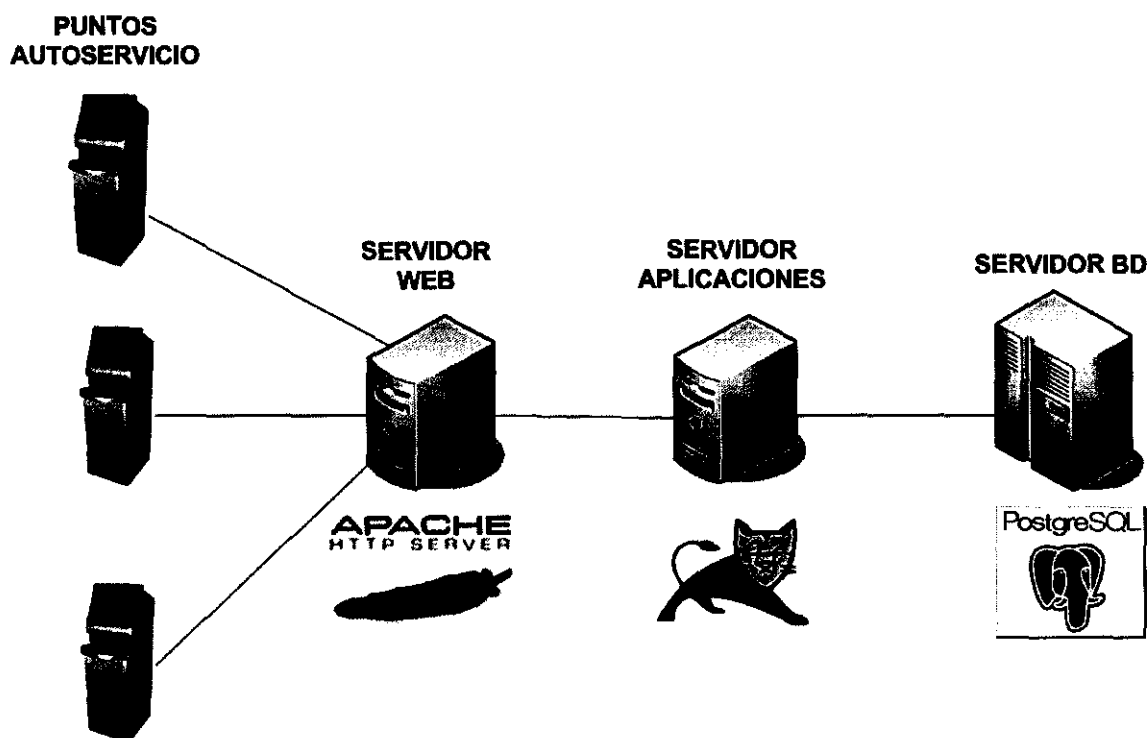
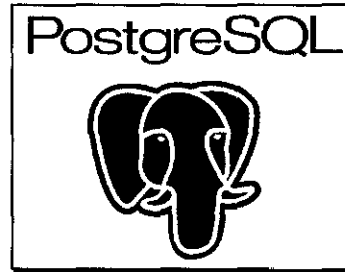


Figura 62. Resumen arquitectura del Sistema

9.1.1 Base de Datos del Sistema



Para realizar una valoración sobre el SGBD más adecuado para el sistema que se va a desarrollar, se decidió apostar en principio por un producto **Open Source**. No obstante, y con el objetivo de que el rendimiento del sistema no se resintiese por esta apuesta, se decidió comparar los SGBD's *Open Source* con el SGBD de Oracle, el motor de Bases de Datos más extendido y por supuesto uno de los más completos. Finalmente, tras analizar con detenimiento todas las posibilidades, se ha determinado utilizar **PostgreSQL** como Sistema Gestor de Bases de Datos. A continuación se muestran una serie de razones que sustentan la decisión tomada:

- Oracle, al ser un producto propietario tenemos que pagar una cantidad sustanciosa de dinero para obtener la licencia.
- Tanto Oracle como **PostgreSQL** son capaces de manipular Bases de Datos de dimensiones enormes, tienen **excelente robustez y escalabilidad** transaccional ya que ambas aplican mecanismos que no están basados en **locks** y disponemos de mecanismos encargados para mantener la integridad de los datos tales como *triggers*.
- MySQL es un SGBD mucho más limitado que Oracle o el mismo Postgresql. De hecho, hasta hace poco carecía de mecanismos tales como *triggers*. Si bien es cierto que en algunas comparativas, se muestra mucho más rápido que Postgresql, esta velocidad en algunos casos está enmascarada por consultas muy simples, de solo lectura y que explotan las optimizaciones específicas de **MySQL**. No obstante, cuando la comparación se realiza en un ambiente transaccional, con moderada concurrencia, usando consultas complejas como las que hay en la vida real, **PostgreSQL** muestra una **abrumadora superioridad**.

Estas afirmaciones se basan en una serie de datos observados en diversas tablas y comparativas publicadas en Internet. A pesar de que es bien sabido por todos que a menudo estas comparativas se ajustan o adecuan con el objetivo de potenciar y resaltar las ventajas de uno de los SGBDs comparado, se ha intentado, en la medida de posible, tomar esta decisión basándose en datos objetivos y constatables, alejándose en todo momento de las comparativas hechas por los propios fabricantes. Entre las fuentes consultadas que sustentan esta decisión cabe destacar las referencias bibliográficas [20], [21], [22] y [23].

9.1.2 Sistema Operativo del Servidor



Siguiendo con la apuesta de utilizar productos **Open Source** y tras realizar un análisis exhaustivo valorando todas las opciones que el mercado ofrece, se ha determinado utilizar el sistema operativo **Linux Ubuntu Server** para los servidores del sistema en base a los siguientes fundamentos:

- **No** se tendrá que pagar una **licencia** como ocurre con otros sistemas tales como Windows.
- Linux Ubuntu Server es un sistema operativo lo suficientemente potente como para cubrir con creces las necesidades del sistema que se pretende desarrollar.
- Desde el punto de vista del soporte, Linux Ubuntu Server dispone de un **destacado servicio de soporte** y una comunidad bastante amplia donde liberan sus versiones finales cada 6 meses, con lo que la detección de *bugs* se soluciona en un tiempo reducido.
- Se trata de un producto *Open Source*, con lo que todos los servicios que albergará podrán contar con estas exigencias.

9.1.3 Servidor Web



Servidor Apache 2.3 [17], es uno de los servidores Web **más utilizados** en Internet, que además proporciona **amplia documentación** y está respaldado por una **comunidad** de desarrollo **brillante**, se sustenta en un amplio número de personas y organizaciones, desde gigantes de la talla de IBM hasta consultores particulares. Dispone un amplio abanico de módulos.

Apache también es también un producto **Open Source** junto con los proyectos que les rodean.

9.1.4 Servidor de Aplicaciones



Se utilizará **Tomcat** [31] como servidor de aplicaciones principalmente por estar mantenido por *Sun System* y *Apache*. Así pues, *Tomcat* se **acopla** a la perfección con el servidor *web* de **Apache**, y con el lenguaje de programación **JAVA**. Una de las funcionalidades principales de *Tomcat* es ser contenedor de *Servlet* y *JSP's*, que permiten una mejor interacción entre la presentación y el dominio.

Otra característica de este servidor de aplicaciones es que proporciona una **buena estabilidad** y un buen soporte para paradigmas *multithread* y para plataformas multiprocesadores.

9.1.5 Datamapper



Se ha decidido utilizar un *Datamapper* para facilitar el mapeo de atributos entre una Base de Datos relacional y el modelo de objetos de la aplicación. De entre todos los productos que el mercado ofrece, se ha determinado utilizar **Hibernate**. Se trata de una herramienta que ha conseguido en un tiempo record una excelente reputación en la comunidad de desarrollo posicionándose claramente como el producto *OpenSource* líder en este campo gracias a sus prestaciones, buena documentación y estabilidad. La elección de **Hibernate** con respecto a otros competidores tales como **iBatis**, **JDO** u **ORM** se ha tomado en base a una serie de motivos tales como [24], [25]:

- Sencillez en el mapeo clases-tablas mediante archivos XML.
- Transparencia con la que realiza este mapeo hasta el punto que el desarrollador no requiere tener conocimientos de SQL.
- Estabilidad.
- Producto *Open Source*.

9.1.6 Framework



Para los sistemas distribuidos, como en el proyecto que estamos tratando, se ha determinado utilizar un **Framework** que permita controlar cada uno de los componentes de que conforman el Sistema. Además, esto permite que la **interacción** entre las diferentes **capas** sea completamente **transparente**.

De entre todos los *Frameworks* que el mercado ofrece actualmente, se ha optado por **Spring** [13] debido a que se trata de un sistema más **ligero y compacto** que algunos de sus competidores como por *EJB*. Se trata de un *Framework Open Source* para el desarrollo de aplicaciones Java que ofrece mucha libertad al desarrollador y una gran cantidad de soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en el desarrollo de software Java. Además se trata de un producto en auge, emergente, que cada vez es más utilizado por las empresas del sector. Cabe destacar también su elevada compatibilidad con proveedores de persistencia [14], [15], [16].

9.1.7 Lenguajes de Programación

Para la programación del sistema, se utilizarán principalmente los lenguajes tales como **JAVA, HTML, CSS** o **JavaScript**, lenguajes *Open Source* ampliamente aceptados por la comunidad Web y que son perfectamente soportables por nuestros servidores. Destacar que el lenguaje principal que predominará en el sistema será el lenguaje Java, como se especifica en el documento de especificaciones técnicas aportado por el cliente.

9.2 Vista Lógica y Física

A continuación, se realiza una descripción detallada de la **Distribución Lógica y Física** de los componentes del Sistema, los **tipos de llamadas** que se realizan entre los servicios, la comunicación entre ellos, así como la interacción y *framework* utilizado para el sistema distribuido.

9.2.1 Descripción Física de los Servidores

- **Servidor de Base de Datos (PostgreSQL)**: Se trata de un **servidor dedicado** exclusivamente para la Base de Datos del sistema.
- **1 servidor Web (Apache)** o capa de presentación del sistema: Será el servidor "más próximo al cliente" que se encargará de aceptar las **peticiones de los clientes**.
- **1 servidor de aplicaciones (Tomcat)** o lógica de negocio: Se trata de un servidor que se encarga del **dominio del Sistema** y el encargado de enlazar con los dos servidores anteriores.

Todos estos servidores serán **servidores dedicados** con lo que el proveedor proporcionará **backups incrementales mensualmente**. Además, se dispondrá de una réplica de cada uno de los servidores en **Akamai** [12]. A continuación se muestra una imagen donde se muestra la relación que se establece entre las diferentes Capas del Sistema y cada uno de los Servidores del mismo.

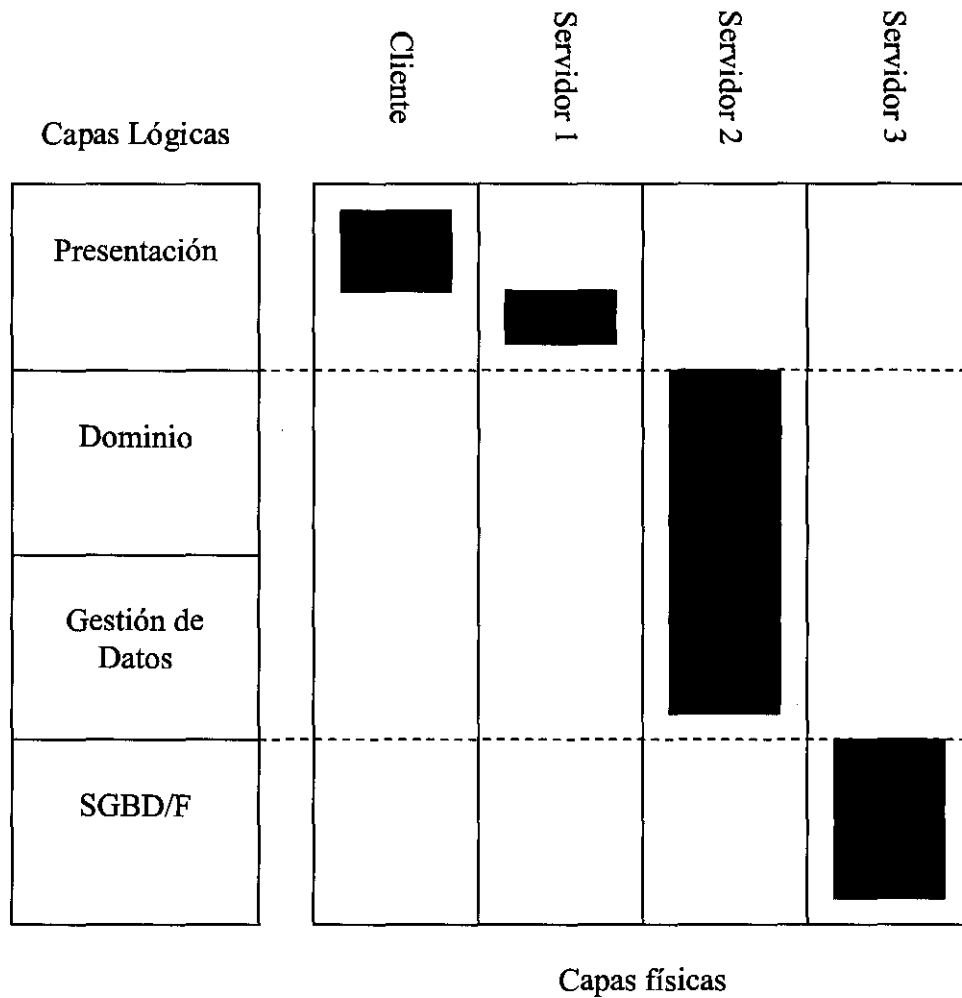


Figura 63. Relación entre las capas del Sistema

9.2.2 Comunicación entre los Servidores (Capas)

A continuación se muestra una imagen que representa de manera esquemática la comunicación e interacción que se establece entre los diferentes Servidores y, por tanto, Capas del Sistema.

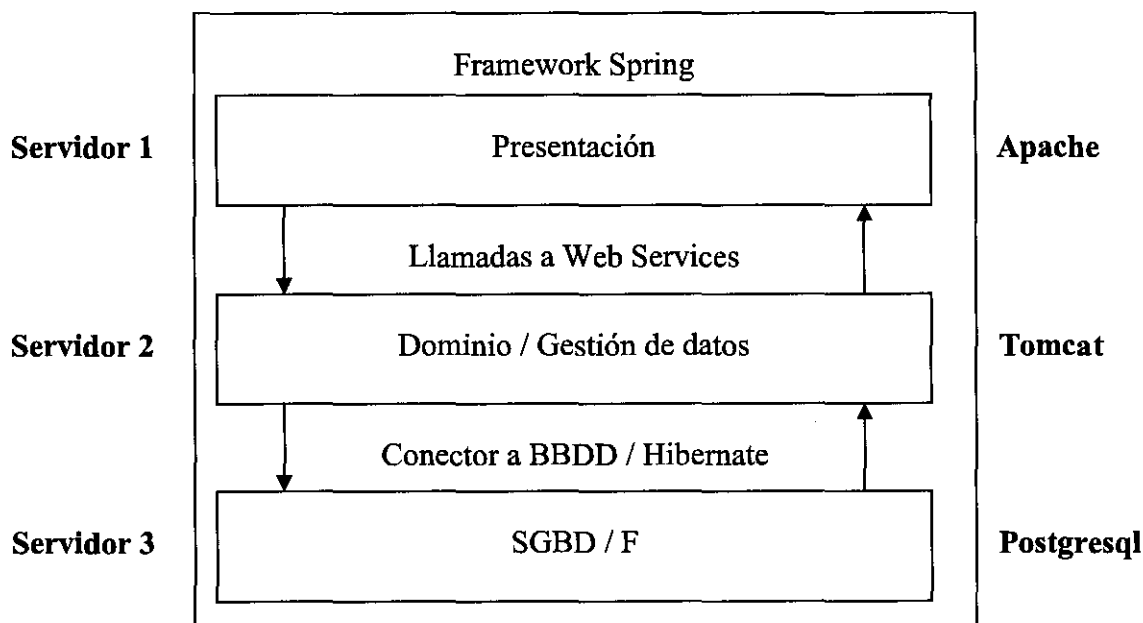


Figura 64. Comunicación entre las capas del Sistema

La comunicación entre la Capa de Presentación y la Capa de Dominio se realizará mediante llamadas a **Web Services**, la implementación de las cuales se realizará mediante **AXIS** y el protocolo **SOAP**.

Por lo que respecta a la comunicación de la Capa de Dominio con **SGBD/F** se ha determinado la utilización de **Hibernate** con el objetivo de que la comunicación entre ambas capas sea totalmente **transparente**. **Hibernate** permite realizar un **Data Mapping** de las clases JAVA que conforman la Capa de Dominio en las tablas relacionales de la Base de Datos.

9.3 Anexos asociados

Se han adjuntado a esta documentación 2 anexos en los que se amplía la información descrita en este apartado sobre las tecnologías y arquitectura del Sistema. En el anexo "**Implementation Model**" se describen minuciosamente cada uno de los componentes que conforman las capas de la arquitectura del Sistema, mientras que en el anexo "**Deploy Model**" se especifica la distribución de cada uno de estos componentes en la infraestructura de redes y máquinas del Sistema.

10 Base de Datos

Como en la mayoría de sistemas de información que se construyen en la actualidad, los Terminales de Autoservicio del SOC están apoyados en una Base de Datos en la que se organiza la información que dichos terminales gestionan y manipulan. Así pues, un adecuado diseño de la Base de Datos del Sistema facilita enormemente el desarrollo y diseño de cada una de las funcionalidades ofrecidas en los Puntos de Autoservicio.

La creación de la Base de Datos del sistema de Puntos de Autoservicio no se limita a la simple definición de las tablas en las que se organizan los datos almacenados, ya que como se ha comentado previamente, esta Base de Datos tiene entre sus objetivos facilitar y optimizar el funcionamiento de cada una de las realizaciones del Sistema. Así pues, se han realizado una serie de tareas encaminadas a optimizar el acceso a los datos almacenados y se han definido un conjunto de mecanismos encargados de velar por la integridad de dichos datos.

10.1 Diagrama Físico de Base de Datos

Partiendo del Diagrama de Clases normalizado del **Analysys Model**, se ha procedido a realizar la traducción de cada una de las clases que conforman el Modelo Conceptual del Sistema en tablas de Base de Datos, definiendo de este modo el **diseño físico de la Base de Datos del Sistema**. Para realizar este diseño físico se ha seguido la premisa fundamental de **evitar** la existencia de **valores nulos** en dichas tablas.

Cabe destacar que se ha determinado que la clase **Fecha** del Modelo Conceptual **no se plasme** en una tabla de la Base de Datos, dado que se ha considerado que la definición de esta clase en el **Analysys Model** tiene como único objetivo facilitar la comprensión del Modelo Conceptual. Así pues, dado que esta clase únicamente representa una serie de fechas asociadas a las inscripciones de los Ciudadanos en servicios, se ha optado por no crear una tabla para representar únicamente un conjunto de fechas.

A continuación se muestra un listado de cada una de las tablas que conforman el **Modelo de Datos** del Sistema. Para cada tabla, se indican las columnas que conforman su **Primary Key** subrayando estas columnas.

- **PuntoAutoservicio** (Id, Oficina, Dirección, Localidad, Comarca, Provincia, Fecha Instalación, Versión)
- **Funcionalidades** (Nombre, Descripción)
- **EstadoFuncionalidad** (IdPunto, NombreFunc, Estado, Fecha Modificación)
 - IdPunto es FK a Punto Autoservicio
 - NombreFunc es FK a Funcionalidades
- **Servicio** (Nombre, Descripción)
- **Documentación** (Nombre, Descripción, Modo Entrega)
- **Servicio Documentación** (NombreServ, NombreDoc)
 - NombreServ es FK a Servicio
 - NombreDoc es FK a Documentación
- **Prestaciones** (Nombre, Descripción, Requisitos)

- **Servicio_Prestaciones** (NombreServ, NombrePrest)
 - NombreServ es FK a Servicio
 - NombrePrest es FK a Prestaciones
- **CiudadanoInscrito** (Id, Password, Logins Erroneos, CuentaBloqueada, Fecha Inscripción, DNI, Nombre, Apellidos, Dirección, Localidad, Comarca, Provincia, Fecha Nacimiento, Teléfono, Email)
- **CiudadanoRegistrado** (Id, Experiencia Profesional, Situación Laboral, Otros Datos Laborales, Formación Regulada, Formación Complementaria, Idiomas, Conocimientos Adicionales, Discapacidades, Colectivo Especial)
 - Id es FK a Ciudadano Inscrito
- **InscripciónServicio** (IdServicio, IdCiudadano, Fecha)
 - IdServicio es FK a Servicio
 - IdCiudadano es FK a Ciudadano Inscrito
- **OfertaEmpleo** (Id, Nombre, Estado, Lugar Recepción, Forma Presentación Candidatura, Número Candidatos, Búsqueda Ocupación/Titulación, Ponderación Perfil, Elaborar Lista Candidatos, Persona Contacto, Fecha Publicación, Fecha Caducidad, Sector, Ocupación Solicitada, Ubicación Empleo, Tipo Contrato, Duración, Sueldo, Horario, Requisitos, Paro, Parámetros Búsqueda, Empresa)
 - Empresa es FK a Empresa Oferta Empleo
- **InscripciónOferta** (IdOferta, IdCiudadano, Fecha)
 - IdOferta es FK a Oferta Empleo
 - IdCiudadano es FK a Ciudadano Registrado
- **Contratación** (Id, Fecha, IdOferta, IdCiudadano)
 - IdOferta, IdCiudadano es FK a Inscripción Oferta
- **Informe** (Id, Fecha, Tipo, Ciudadano)
 - Tipo es FK a Tipo Informe
 - Ciudadano es FK a Ciudadano Inscrito
- **DatoInforme** (Informe, Campo, Valor)
 - Informe es FK a Informe
- **TipoInforme** (Nombre, Descripción)
- **CampoInforme** (TipoInforme, Campo, Campo Cuenta)
 - TipoInforme es FK a Tipo Informe
- **Solicitud** (Id, Estado, Fecha, Tipo)
 - Tipo es FK a Tipo Solicitud
- **CiudadanoInscrito_Solicitud** (IdSolicitud, IdCiudadano)
 - IdSolicitud es FK a Solicitud
 - IdCiudadano es FK a Ciudadano Inscrito
- **DatoSolicitud** (Solicitud, Campo, Valor)
 - Solicitud es FK a Solicitud
- **TipoSolicitud** (Nombre, Descripción)
- **CampoSolicitud** (TipoSolicitud, Campo, Campo Cuenta)
 - TipoSolicitud es FK a Tipo Solicitud
- **Demanda** (Id, Fecha Renovación, Fecha Caducidad, Clave Renovación, Sector, Ocupaciones Solicitadas, Bolsa de Centro, Disponibilidad, Disponibilidad Fuera Comunidad, Ámbito, Estado)
 - Estado es FK a Estado
- **Estado** (Nombre, Descripción)
- **PerfilDemanda** (Ciudadano, Demanda)
 - Ciudadano es FK a Ciudadano Inscrito
 - Demanda es FK a Demanda
- **CursoFormación** (Id, Nombre, Fecha Inicio, Fecha Fin, Estado, Lugar, Horario, Descripción, Requisitos Acceso, Conocimientos Adquiridos, Coste, Empresa, Sector, Temario)
 - Empresa es FK a Empresa Curso
- **InscripciónCurso** (Ciudadano, Curso, Fecha)
 - Ciudadano es FK a Ciudadano Inscrito
 - Curso es FK a Curso Formación
- **PerfilDemanda_Curso** (PerfilDemanda, Curso)
 - PerfilDemanda es FK a Perfil Demanda

- Curso es FK a Curso Formación
- **PerfilDemanda_Oferta** (PerfilDemanda, Oferta)
 - PerfilDemanda es FK a Perfil Demanda
 - Oferta es FK a Oferta Empleo
- **Empresa** (CIF, Nombre, Dirección, Localidad, Comarca, Provincia, Email, Teléfono)
- **EmpresaOfertaEmpleo** (Empresa, Actividad Laboral, Contrataciones)
 - Empresa es FK a Empresa
- **EmpresaCurso** (Empresa, TipoCursos, PorcentajeAprobados, Privada)
 - Empresa es FK a Empresa
- **SolucionesSoporte** (Id, Nombre, Pantalla, Descripción)

A continuación, podemos observar un diagrama que representa el **Modelo Físico** de la Base de Datos del Sistema. Este diagrama se ha construido en base al listado de las tablas definido.

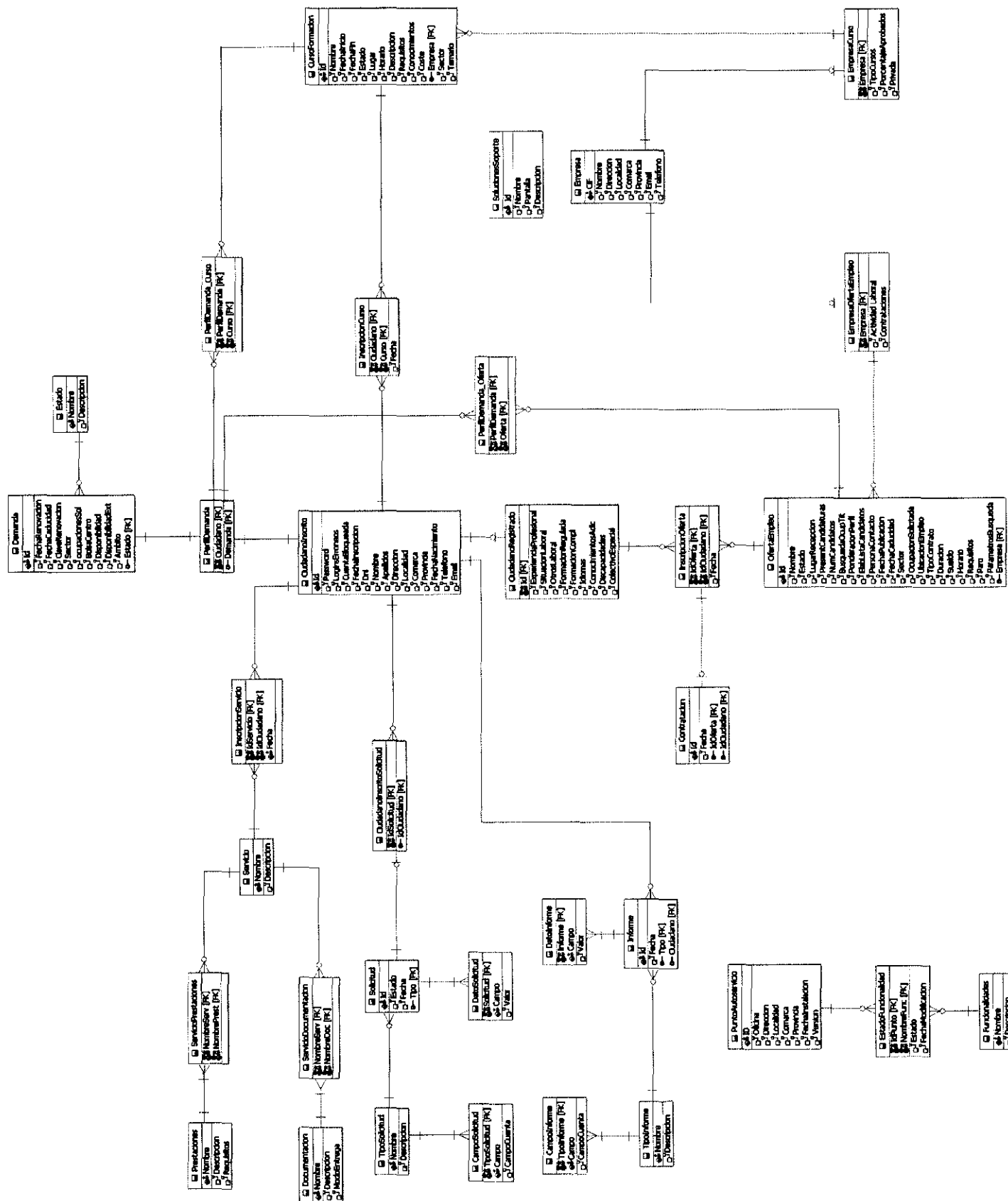


Figura 65. Diagrama físico de Base de Datos

10.2 Optimización de Consultas: Estudio de Índices

Con el fin de **optimizar** las consultas que se realizan sobre la Base de Datos del Sistema, se ha realizado un estudio para determinar que **índices** se deberían definir con el fin de facilitar, agilizar y optimizar el acceso de a los datos. Así pues, a continuación se muestran para cada uno de los Casos de Uso del Sistema de Puntos de Autoservicio, las consultas que más relevantes que se realizan en el transcurso de sus realizaciones:

Gestión Cuenta y Demanda:

Activar Demanda:

- `SELECT d.* FROM Demanda d, PerfilDemanda pd WHERE pd.Ciudadano=:ciudadano AND d.Id=pd.Demanda;`
- `SELECT * FROM Demanda WHERE Id=:deman;`
- `SELECT * FROM Estado WHERE Nombre=:tipoEstado;`

Actualizar Perfil:

- `SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;`
- `SELECT * FROM CiudadanoRegistrado WHERE Id=:ciudadano;`
- `SELECT * FROM PerfilDemanda WHERE Ciudadano=:ciudadano;`
- `SELECT d.* FROM Demanda d, PerfilDemanda pd WHERE pd.Ciudadano=:ciudadano AND d.Id=pd.Demanda;`

Eliminar Cuenta

- `SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;`
- `SELECT * FROM InscpcionServicio WHERE IdCiudadano=:ciudadano;`
- `SELECT * FROM InscpcionOferta WHERE IdCiudadano=:ciudadano;`
- `SELECT * FROM InscpcionCurso WHERE Ciudadano=:ciudadano;`
- `SELECT * FROM Informe WHERE Ciudadano=:ciudadano;`
- `SELECT * FROM CiudadanoInscrito_Solicitud WHERE IdCiudadano=:ciudadano;`

Crear Cuenta/Inscripción Abreviada

- `SELECT * FROM Estado WHERE Nombre=:tipoEstado;`
- `SELECT * FROM CursoFormacion WHERE Sector=:sector;`
- `SELECT * FROM OfertaEmpleo WHERE Sector=:sector;`
- `SELECT Nombre FROM TipoInforme;`
- `SELECT * FROM CampoInforme WHERE TipoInforme=:tipo;`

Renovar Demanda

- `SELECT d.* FROM Demanda d, PerfilDemanda pd WHERE pd.Ciudadano=:ciudadano AND d.Id=pd.Demanda;`
- `SELECT * FROM Estado WHERE Nombre=:tipoEstado;`



- SELECT * FROM Informe WHERE Ciudadano=:ciudadano AND Tipo=:tipoInforme;
- SELECT * FROM DatosInforme WHERE Informe=:inf AND Campo=:campo;

Gestión Información:

Acceso Información:

- SELECT Nombre FROM Servicio;
- SELECT * FROM Servicio WHERE Nombre=:serv;
- SELECT * FROM Servicio_Documentacion WHERE NombreServ=:serv;
- SELECT * FROM Servicio_Prestaciones WHERE NombreServ=:serv;

Rellenar Solicitud:

- SELECT Nombre FROM TipoSolicitud;
- SELECT * FROM TipoSolicitud WHERE Nombre=:tipo;
- SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;
- SELECT * FROM CampoSolicitud WHERE TipoSolicitud=:tipo;

Petición Informe:

- SELECT Tipo FROM Informe WHERE Ciudadano=:ciudadano;
- SELECT * FROM Informe WHERE Ciudadano=:ciudadano AND Tipo=:tipoInforme;
- SELECT * FROM DatosInforme WHERE Informe=:infor;

Gestión Formación:

Buscar Cursos:

- SELECT * FROM CursoFormación WHERE Id=:curso;
- SELECT * FROM CursoFormacion WHERE Nombre=:nom;
- SELECT * FROM CursoFormacion WHERE Lugar=:dir;
- SELECT * FROM CursoFormacion WHERE Temario=:tem;
- SELECT * FROM CursoFormacion WHERE Empresa=:emp;
- SELECT * FROM CursoFormacion WHERE FechaInicio=:fecha;
- SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;
- SELECT * FROM PerfilDemanda WHERE Ciudadano=:ciudadano;
- SELECT c.* FROM PerfilDemanda_Curso pdc, CursoFormacion c WHERE pdc.PerfilDemanda=:perfil AND pdc.Curso=c.Id

Inscripción Curso Formación:

- SELECT * FROM CursoFormación WHERE Id=:curso;
- SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;
- SELECT * FROM Empresa WHERE CIF=:empresa;

Gestión Ofertas:

Buscar Ofertas:

- SELECT * FROM OfertaEmpleo WHERE Id=:oferta;
- SELECT * FROM OfertaEmpleo WHERE Nombre=:nom;
- SELECT * FROM OfertaEmpleo WHERE UbicacionEmpleo=:ubiEmpleo;
- SELECT * FROM OfertaEmpleo WHERE Sector=:sector;
- SELECT * FROM OfertaEmpleo WHERE Empresa=:emp;
- SELECT * FROM OfertaEmpleo WHERE OcupacionSolicitada=:ocupSol;
- SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;
- SELECT * FROM PerfilDemanda WHERE Ciudadano=:ciudadano;
- SELECT o.* FROM PerfilDemanda_Oferta pdo, OfertaEmpleo o WHERE pdo.PerfilDemanda=:perfil AND pdo.Oferta=o.Id

Inscripción Oferta Empleo:

- SELECT * FROM OfertaEmpleo WHERE Id=:oferta;
- SELECT * FROM CiudadanoRegistrado WHERE Id=:ciudadano;
- SELECT * FROM Empresa WHERE CIF=:empresa;

Gestión Sesión:

Login:

- SELECT * FROM CiudadanoInscrito WHERE Id=:ciudadano;

Soporte y Atención al Cliente:

Petición Soporte:

- SELECT * FROM SolucionesSoporte WHERE Pantalla=:pant AND Nombre LIKE :descripción

Gestión Punto Autoservicio:

Activar Servicio/Desactivar Servicio:

- SELECT Nombre FROM Funcionalidades;
- SELECT * FROM Funcionalidades WHERE Nombre=:func;
- SELECT * FROM EstadoFuncionalidad WHERE NombreFunc=:func;
- SELECT * FROM PuntoAutoservicio WHERE Id=:punto;
- SELECT p.* FROM PuntoAutoservicio p, EstadoFuncionalidad e WHERE p.Id=:punto AND e.NombreFunc=:func AND e.Estado=:estado AND p.Id=e.IdPunto;

- SELECT p.* FROM PuntoAutoservicio p, EstadoFuncionalidad e WHERE p.Oficina=:oficina AND e.NombreFunc=:func AND e.Estado=:estado AND p.Id=e.IdPunto;
- SELECT p.* FROM PuntoAutoservicio p, EstadoFuncionalidad e WHERE p.Localidad=:loc AND e.NombreFunc=:func AND e.Estado=:estado AND p.Id=e.IdPunto;
- SELECT p.* FROM PuntoAutoservicio p, EstadoFuncionalidad e WHERE p.Comarca=:com AND e.NombreFunc=:func AND e.Estado=:estado AND p.Id=e.IdPunto;
- SELECT p.* FROM PuntoAutoservicio p, EstadoFuncionalidad e WHERE p.Provincia=:prov AND e.NombreFunc=:func AND e.Estado=:estado AND p.Id=e.IdPunto;
- SELECT p.* FROM PuntoAutoservicio p, EstadoFuncionalidad e WHERE p.Version=:vers AND e.NombreFunc=:func AND e.Estado=:estado AND p.Id=e.IdPunto;
- SELECT * FROM EstadoFuncionalidad WHERE IdPunto=:punto AND NombreFunc=:func;

Tras revisar el conjunto anterior de consultas, se puede constatar que en la mayoría de ellas, las columnas de las tablas que aparecen en la cláusula *WHERE* son o bien *Primary Keys* o bien *Foreign Keys*. Considerando que a día de hoy, la mayoría, por no decir todos los Sistemas Gestores de Bases de Datos (SGBDs) del mercado **crean automáticamente** índices para las *Primary Keys* y las *Foreign Keys* de las tablas, no se incluirá la creación de estos índices en el Modelo de Datos del Sistema de Puntos de Autoservicio, dado que se puede considerar que el SGBD los creará automáticamente.

Por otra parte cabe destacar que en la cláusula *WHERE* de algunas de las consultas a la Base de Datos listadas anteriormente aparecen columnas tales como la columna *Estado* de la tabla *EstadoFuncionalidad*. El **número de valores diferentes** que pueden tomar estas columnas es tan limitado que la definición de un índice ella no optimizaría en absoluto la consulta sobre la Base de Datos.

Así pues, la mayoría de los índices que se definirán están asociados a los **filtros de búsqueda** de Cursos de Formación y Ofertas de Empleo. Se trata de 2 realizaciones que se realizarán con mucha frecuencia, dado que desempeñan un papel fundamental en el objetivo fundamental del Sistema de Puntos de Autoservicio: la incorporación de los Ciudadanos al mercado laboral. Destacar también la creación de una serie de índices asociadas a tareas de administración y gestión de los Puntos de Autoservicio. Por tanto, los índices que se crearán con el fin de optimizar las consultas a la Base de Datos del Sistema son los siguientes:

1. Tabla *CursoFormación* → Columna *FechaInicio*
2. Tabla *CursoFormación* → Columna *Lugar*
3. Tabla *CursoFormación* → Columna *Nombre*
4. Tabla *CursoFormación* → Columna *Sector*
5. Tabla *CursoFormación* → Columna *Temario*
6. Tabla *OfertaEmpleo* → Columna *Nombre*
7. Tabla *OfertaEmpleo* → Columna *OcupacionSolicitada*
8. Tabla *OfertaEmpleo* → Columna *Sector*
9. Tabla *OfertaEmpleo* → Columna *UbicacionEmpleo*
10. Tabla *PuntoAutoservicio* → Columna *Comarca*
11. Tabla *PuntoAutoservicio* → Columna *Localidad*

- 12. Tabla **PuntoAutoservicio** → Columna **Provincia**
- 13. Tabla **PuntoAutoservicio** → Columna **Version**
- 14. Tabla **SolucionesSoporte** → Columna **Pantalla**

En el anexo “**Scripts de Base de Datos**” se puede consultar el código asociado a la definición de estos índices en la Base de Datos del Sistema.

10.3 Integridad del Modelo de Datos

Con el fin de preservar la **integridad** de la Base de Datos definida anteriormente, se ha procedido a definir una serie de disparadores o **triggers** encargados de mantener y velar por el cumplimiento de la integridad del modelo. En el apartado asociado al **Design Model** ya se han hecho diversas referencias a estos **triggers** debido a que la definición de estos disparadores ha facilitado considerablemente el diseño de algunas realizaciones de los Casos de Uso de Sistema, evitar en ciertos casos la aplicación de ciertos Patrones de Diseño tales como el **Patrón Observador** para resolver una serie de necesidades. Así pues, dada la estrecha relación que se establece entre los **triggers** definidos y ciertas realizaciones, se describirán a continuación cada uno de estos **triggers** indicando para cada uno de ellos la realización que ha provocado su creación. En caso de que se desee consultar el código PL/SQL asociado a la definición de estos **triggers**, se ha adjuntado dicho código en el anexo “**Scripts de Base de Datos**”. Destacar que el código de estos **triggers** se ha definido en función del Sistema Gestor de Bases de Datos escogido, es decir, **PostgreSQL**.

10.3.1 Trigger realización Eliminar Cuenta

En la realización **Eliminar Cuenta** del Caso de Uso Gestión Cuenta y Demanda, se produce una eliminación de los datos de las tablas **CiudadanoInscrito** y **CiudadanoRegistrado**, es decir, los datos asociados a la cuenta del Ciudadano en el Sistema. Pero como se puede observar en el Modelo Conceptual del Sistema, existen una serie de datos almacenados en otras tablas de la Base de Datos cuya existencia está estrechamente vinculada a los datos de una cuenta. Así pues, la eliminación de los datos de una cuenta debe conllevar la eliminación de este conjunto de **datos relacionados** con ella. Este conjunto de datos comprende:

- Inscripciones en Servicios.
- Inscripciones en Cursos de Formación.
- Inscripciones en Ofertas de Empleo.
- Informes del Ciudadano.
- Solicitudes del Ciudadano.
- Datos asociados a la Demanda del Ciudadano.

Así pues, el **trigger** definido deberá ejecutarse cuando se produzca una operación de **borrado** en la tabla **CiudadanoInscrito** y deberá eliminar toda la información asociada al Ciudadano indicada anteriormente.

10.3.2 Triggers realización Actualizar Perfil

Como se ha comentado anteriormente, durante el proceso de especificación del Sistema se decidió materializar los datos que conforman los Informes de un Ciudadano. Dado que la mayoría de estos datos se derivan de la información asociada a la cuenta del Ciudadano, una **actualización** en los datos de la **cuenta** debe comportar una **actualización** de los datos de los **Informes** con el fin de preservar la integridad del modelo.

Así pues, se deberá definir un *trigger* para cada una de las tablas que se ven afectadas por la realización **Actualizar Perfil**, es decir, las tablas **CiudadanoInscrito**, **CiudadanoRegistrado** y **Demanda**, con el objetivo de que actualicen los datos de los Informes del Ciudadano que modifica su perfil. Cada uno de estos disparadores se ejecutará cada vez que se produzca una operación de modificación de los datos de la tabla a la que está asociado.

10.4 Implementación Física de la Base de Datos

Una vez realizado el diseño del Modelo de Datos del Sistema se ha procedido a implementar dicho diseño en el Sistema Gestor de Bases de Datos seleccionado: **PostgreSQL 8.3**. La decisión de realizar esta implementación radica en que, como se ha comentado al inicio de esta documentación, uno de los objetivos fundamentales de este proyecto es proporcionar no sólo una base sobre la que sustentar una futura implementación de los Puntos de Autoservicio, sino que se pretende iniciar la implementación de dicho Sistema, proporcionando una implementación física tanto para la Capa de Presentación como para la Capa de Persistencia.

Como ya se ha comentado previamente, existe un anexo en esta documentación, **"Scripts de Base de Datos"**, en el que se encuentra el código asociado a la definición de cada uno de los objetos que determinan la Base de Datos del Sistema: **Tablas, Índices y Triggers**.

Por otra parte, cabe destacar también que en el CD adjuntado a esta documentación se ha incluido la exportación realizada de la propia Base de Datos del Sistema.

11 Planificación y Gestión del Proyecto

En cualquier proyecto de cierta envergadura resulta altamente recomendable realizar una planificación pormenorizada de cada una de las tareas que se llevarán a cabo durante el desarrollo del proyecto, así como una gestión de los diferentes recursos que se utilizarán en dicho desarrollo.

Hay que remarcar que en este apartado se incluye únicamente un breve resumen del conjunto de acciones y tareas asociadas a la planificación y gestión del proyecto que se han realizado. No obstante, en el anexo **“Software Development Plan”** se incluye una descripción detallada de estas acciones, especificando no únicamente una planificación temporal y evaluación económica del proyecto, sino también una descripción de la organización estructural de dicho proyecto, una **evaluación de los riesgos** del mismo y una **planificación** de las fases asociadas a la **construcción** del Sistema.

Así pues, retomando la breve descripción de estas tareas de planificación, cabe destacar que, siguiendo las bases establecidas en la metodología RUP, la realización de este proyecto se ha dividido en una serie de fases o iteraciones en las cuales se agrupan una serie de tareas que presentan ciertas características comunes. Así pues, siguiendo las recomendaciones establecidas en la metodología RUP, este proyecto se ha dividido en 3 fases principales:

- **Inicio**
- **Elaboración 1**
- **Elaboración 2**

La fase de **Inicio** o **Inception** comprende una serie de tareas asociadas al análisis pormenorizado del problema y los requisitos que debe cumplir el Sistema que se pretende desarrollar. Así pues, en esta fase se realizarán una serie de tareas entre las que cabe destacar:

- Definición del problema.
- Definición de los requisitos funcionales y no funcionales del Sistema.
- Definición del Modelo de Negocio del SOC.
- Descripción de los Casos de Uso del Sistema.

La fase de **Elaboración 1** esta asociada principalmente a tareas de especificación y análisis del Sistema. Cabe remarcar que en esta fase se realizará también el diseño e implementación de la interficie gráfica de usuario. Así pues, durante esta iteración se realizarán un conjunto de tareas entre las que destacan:

- Definición de Modelo de Casos de Uso del Sistema.
- Diseño y creación de la interficie gráfica de usuario.
- Definición del Modelo Conceptual del Sistema.
- Definición del Modelo de Análisis del Sistema.

Finalmente, la fase de **Elaboración 2** comprende principalmente un conjunto de tareas relacionadas con el diseño de las funcionalidades del Sistema y la definición

de la Base de Datos del Sistema. Del mismo modo, cabe remarcar la existencia de un conjunto de actividades asociadas a la definición de la arquitectura física del Sistema. Así pues, durante esta fase se llevarán a cabo principalmente las siguientes tareas:

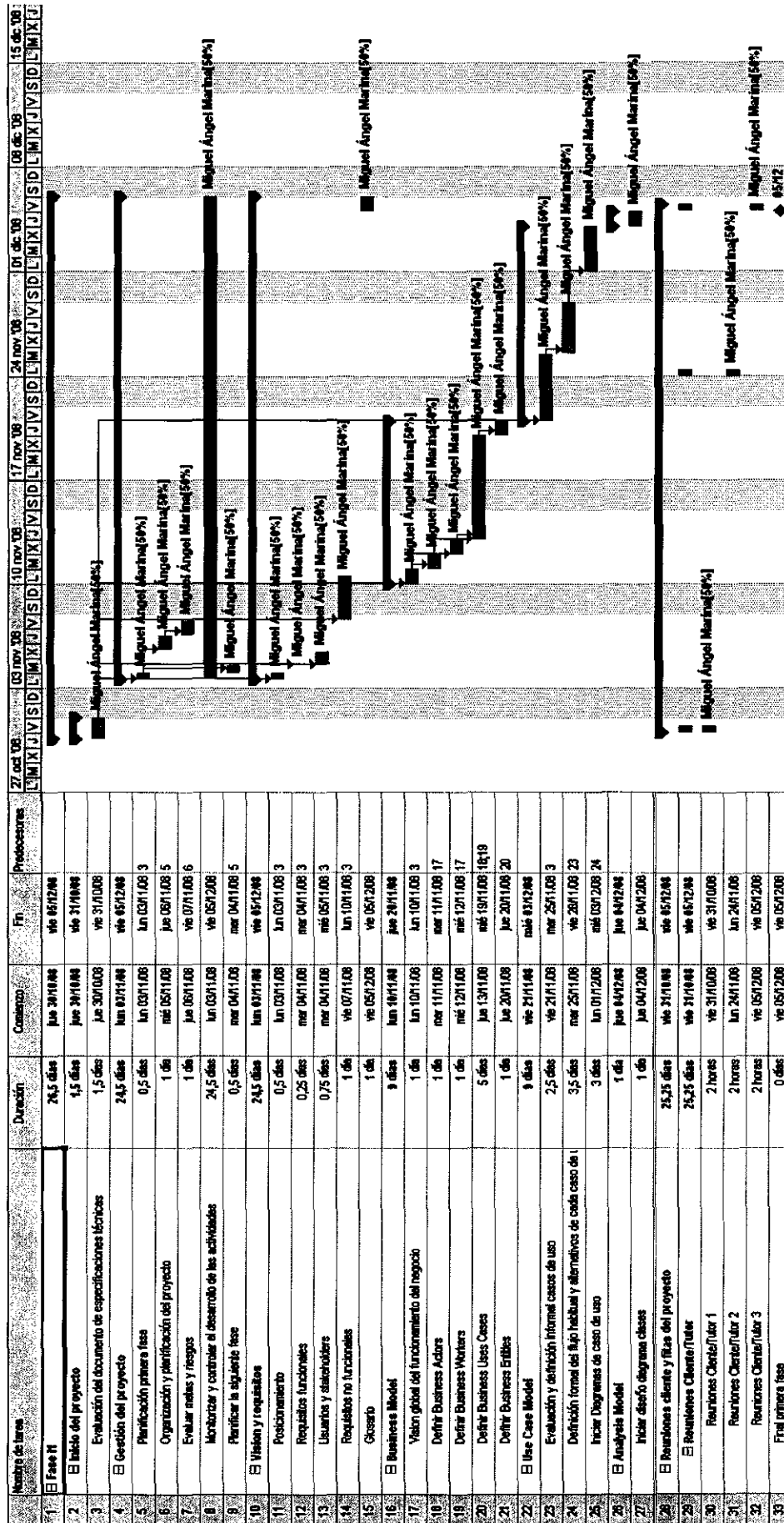
- Diseño de cada una de las funcionalidades del Sistema.
- Definición de la arquitectura física del Sistema.
- Definición de los modelos de despliegue e implementación del Sistema.
- Diseño e implementación de la Base de Datos del Sistema.
- Definición del plan de pruebas y de los casos de prueba.

11.1 Planificación Temporal

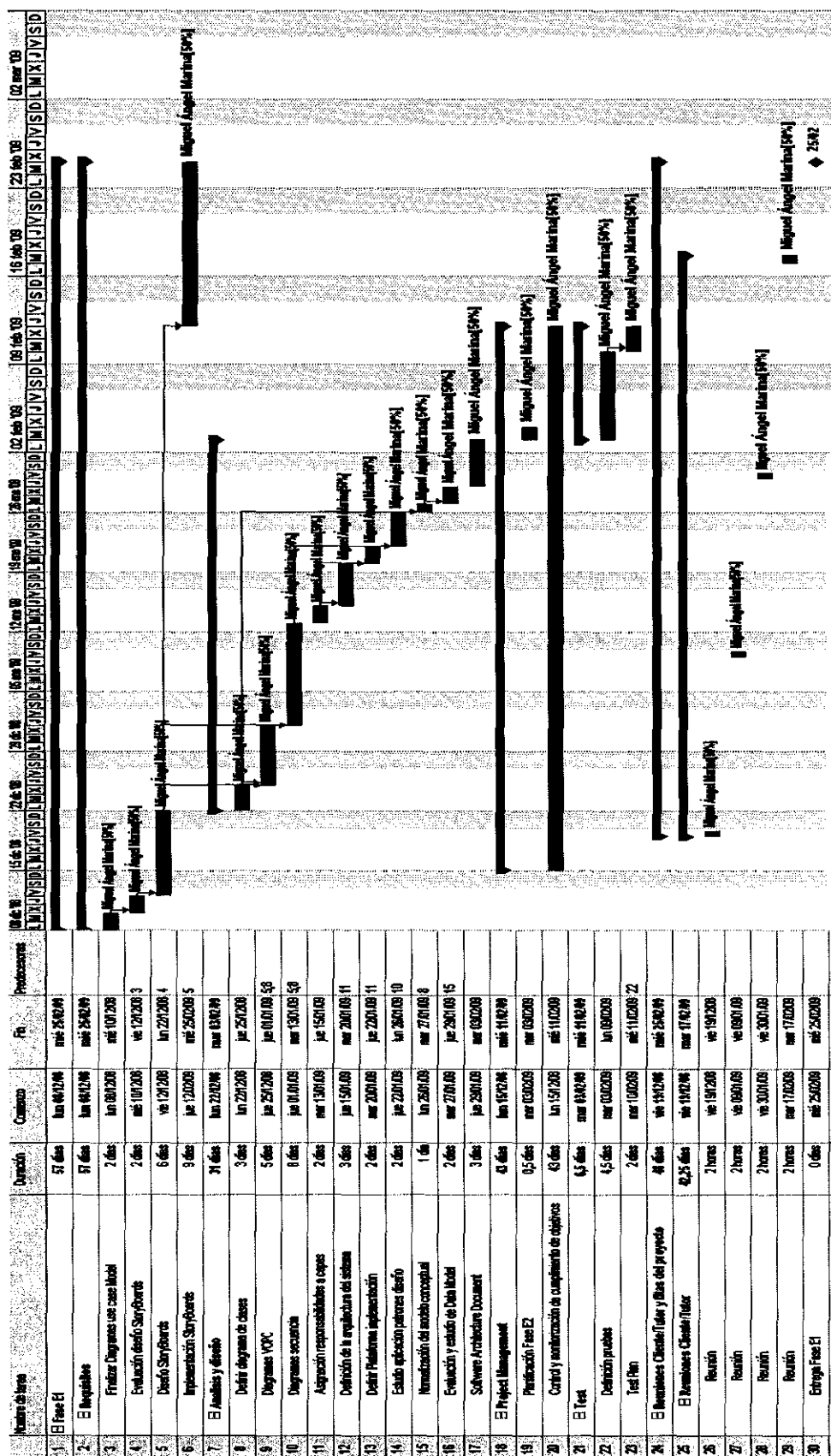
Para facilitar la comprensión de la planificación de cada una de las fases en las que se ha dividido el desarrollo del proyecto, cabe destacar que esta planificación se ha realizado considerando una dedicación de 4 horas diarias, 5 días a la semana.

Por otra parte, cabe remarcar que la estimación del esfuerzo destinado a cada una de las tareas se ha realizado en base a mi propia experiencia. Se ha optado por esta vía como consecuencia de mi destacada participación durante más de un año en el Proyecto KIOSK de la empresa Mitsubishi Electric, proyecto de similares características al que nos atañe, ya que este Proyecto de la empresa nipona se fundamenta en una serie de Terminales de Autoservicio que permiten al Ciudadano gestionar, manipular e imprimir fotografías digitalizadas.

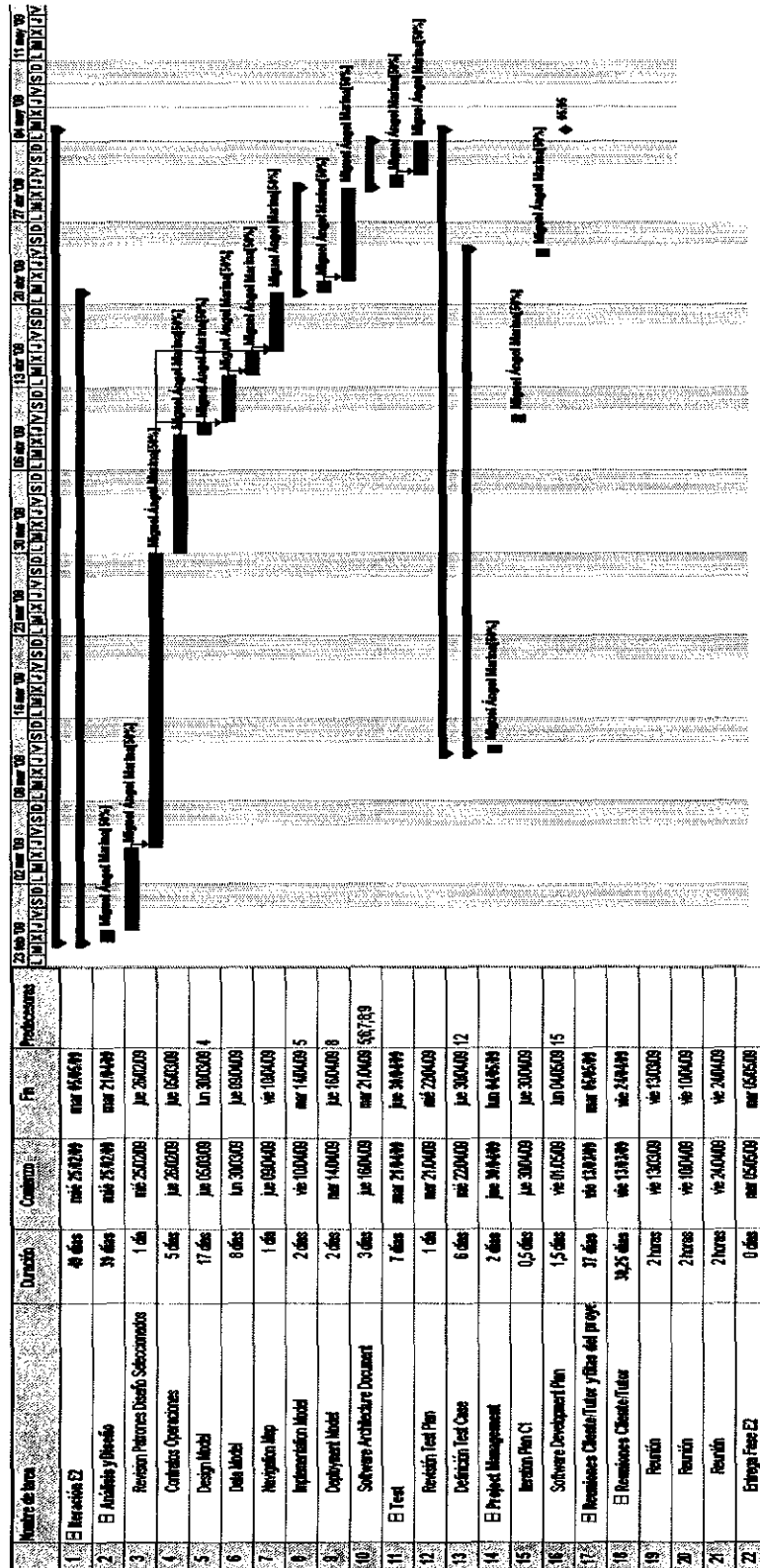
A continuación se muestra la planificación temporal de la fase de **Inicio** mediante un Diagrama de Gantt. Esta iteración tiene como fecha inicial el 30 de octubre de 2008 y como fecha de finalización el día 5 de diciembre del mismo año.



Del mismo modo, en la imagen siguiente podemos observar el Diagrama de Gantt correspondiente a la iteración **Elaboración 1**. Esta fase tiene como fecha de inicio el 8 de diciembre de 2008 y como fecha de finalización el 25 de febrero del 2009.



Finalmente, se incluye el Diagrama de Gantt de la fase **Elaboración 2**, la cual se inicia el 25 de febrero de 2009 y finaliza el 5 de mayo del mismo año.



11.2 Coste Económico

Una vez realizada la planificación temporal del proyecto, conviene realizar una aproximación al coste económico del desarrollo del Sistema en base a la dedicación temporal y de recursos estimada en dicha planificación. Para facilitar la comprensión del coste económico del proyecto, se realizará un desglose del mismo en base a las diferentes tareas que componen cada uno de las fases del proyecto. No obstante, considerando que este Proyecto de Final de Carrera establece las bases de una futura implementación física de los Puntos de Autoservicio del *Servei d'Ocupació de Catalunya*, se realizará una evaluación no únicamente del coste asociado a este Proyecto de Final de Carrera, sino de todo el proceso de construcción del Sistema, incluyendo tanto las fases análisis y diseño del mismo contempladas en este proyecto, como las posteriores etapas de construcción y transición. La planificación temporal de estas etapas de construcción y transición puede consultarse en el anexo "**Software Development Plan**".

A continuación se muestra una tabla donde se especifica el coste económico de cada una de las tareas realizadas durante las diferentes fases del proyecto. Para cada tarea se indica:

- Nombre de la tarea.
- Horas dedicadas a la tarea.
- Recursos o personal dedicado a la tarea.
- Coste unitario del personal dedicado.
- Coste total de la tarea.

Cabe remarcar que en base a los costes estipulados en el mercado actual, se han considerado los siguientes costes unitarios:

- Coste Jefe de Proyecto: 60 €
- Coste Analista: 55 €
- Coste Arquitecto: 55 €
- Coste Programador: 45 €

Fase Inicio				
Tarea	Horas	Recursos	Coste Unitario	Coste
Evaluación documento cliente	6	1	60 €	360 €
Definición del problema	6	1	60 €	360 €
Análisis de requisitos	8	1	55 €	440 €
Modelo de Negocio	36	1	55 €	1.980 €
Gestión y Planificación	12	1	60 €	720 €
Descripción Casos de Uso	24	1	55 €	1.320 €
Modelo Casos de Uso	12	1	55 €	660 €
Glosario	4	1	55 €	220 €

Coste Total Fase de Inicio				6.060 €
Fase Elaboración 1				
Tarea	Horas	Recursos	Coste Unitario	Coste
Modelo Casos de Uso	8	1	55 €	440 €
Modelo de Análisis	84	1	55 €	4.620 €
<i>Storyboards</i>	68	1	55 €	3.740 €
Arquitectura del Sistema	20	1	55 €	1.100 €
Gestión y Planificación	8	1	60 €	480 €
Modelo de Diseño	12	1	55 €	660 €
Modelo de Datos	8	1	55 €	440 €
Definición pruebas	26	1	60 €	1.560 €
Coste Total Fase de Elaboración 1				13.040 €
Fase Elaboración 2				
Tarea	Horas	Recursos	Coste Unitario	Coste
Modelo de Diseño	92	1	55 €	5.060 €
Modelo de Datos	32	1	55 €	1.760 €
<i>Navigation Map</i>	4	1	55 €	220 €
Modelo de Implementación	8	1	55 €	440 €
Modelo de Despliegue	8	1	55 €	440 €
Definición pruebas	28	1	60 €	1.680 €
Gestión y Planificación	12	1	60 €	720 €
Coste Total Fase de Elaboración 2				10.320 €
Fase Construcción 1				
Tarea	Horas	Recursos	Coste Unitario	Coste
Implementación física BD	76	1	55 €	4.180 €
Acceso Información	48	4	45 €	8.640 €
Rellenar Solicitud	64	4	45 €	11.520 €
Buscar Cursos	80	4	45 €	14.400 €
Buscar Ofertas	72	4	45 €	12.960 €
Activar Servicio	88	4	45 €	15.840 €
Desactivar Servicio	64	4	45 €	11.520 €
Petición Soporte	48	4	45 €	8.640 €
Imprimir	40	4	45 €	7.200 €
Test Base de Datos	40	1	45 €	1800 €
Test Acceso Información	24	1	45 €	1080 €
Test Rellenar Solicitud	24	1	45 €	1080 €
Test Buscar Cursos	24	1	45 €	1080 €



Test Buscar Ofertas	24	1	45 €	1080 €
Test Activar Servicio	24	1	45 €	1080 €
Test Desactivar Servicio	24	1	45 €	1080 €
Test Petición Soporte	32	1	45 €	1440 €
Test Imprimir	16	1	45 €	720 €
Gestión y Planificación	24	1	60 €	1440 €
Coste Total Fase de Construcción 1				106.780 €

Fase Transición 1

Tarea	Horas	Recursos	Coste Unitario	Coste
Gestión y Planificación	4	1	60 €	240 €
Manual Usuario	64	1	45 €	2880 €
Documentación técnica	80	1	45 €	3600 €
Documentación mantenimiento	48	1	45 €	2160 €
Montar servidores	32	1	45 €	1440 €
Instalar <i>software</i> terminales	80	1	45 €	3600 €
Formación mantenimiento	56	1	45 €	2520 €
Formación guías	48	1	45 €	2160 €
Coste Total Fase de Transición 1				18600 €

Fase Construcción 2

Tarea	Horas	Recursos	Coste Unitario	Coste
Modificar BD y Carga de Datos	88	1	45 €	3.960 €
Login	40	3	45 €	5.400 €
Logout	32	3	45 €	4.320 €
Rellenar Solicitud	24	4	45 €	4.320 €
Petición Informes	64	4	45 €	11.520 €
Buscar Cursos	24	2	45 €	2.160 €
Inscripción Cursos	56	2	45 €	5.040 €
Buscar Ofertas	24	2	45 €	2.160 €
Inscripción Ofertas	64	2	45 €	5.760 €
Revisión funcionalidades C1	144	1	45 €	6.480 €
Test Base de Datos	72	1	45 €	3.240 €
Test Login	24	1	45 €	1.080 €
Test Logout	16	1	45 €	720 €
Test Rellenar Solicitud	16	1	45 €	720 €
Test Petición Informes	24	1	45 €	1.080 €
Test Buscar Cursos	16	1	45 €	720 €
Test Inscripción Cursos	24	1	45 €	1.080 €

Test Buscar Ofertas	16	1	45 €	720 €
Test Inscripción Ofertas	24	1	45 €	1.080 €
Test funcionalidades C1	64	1	45 €	2.880 €
Gestión y Planificación	24	1	60 €	1.440 €
Coste Total Fase de Construcción 2				65.880 €
Fase Transición 2				
Tarea	Horas	Recursos	Coste Unitario	Coste
Gestión y Planificación	4	1	60 €	240 €
Manual Usuario	56	1	45 €	2.520 €
Documentación técnica	64	1	45 €	2.880 €
Documentación mantenimiento	40	1	45 €	1.800 €
Montar servidores	40	1	45 €	1.800 €
Instalar <i>software</i> terminales	104	1	45 €	4.680 €
Formación mantenimiento	48	1	45 €	2.160 €
Formación guías	48	1	45 €	2.160 €
Coste Total Fase de Transición 2				18.240 €

Así pues, en base a esta serie de cifras y considerando que este **Proyecto Final de Carrera** comprende las fases de Inicio, Elaboración 1, Elaboración 2 y la tarea de implementación física de la Base de Datos, se puede concluir que el **coste** de este Proyecto es de: **39.740 €**. Cabe destacar que se ha considerado el coste asociado a la implementación física de la Base de Datos de la fase C1, ya que las tareas asociadas a la Base de Datos en la fase de construcción C2 están estrechamente ligadas a la carga de datos reales, carga que no se ha realizado en este Proyecto.

Sin embargo, estas cifras no permiten calcular únicamente el coste económico de este Proyecto de Final de Carrera, sino el **coste total** del proceso de construcción del Sistema de Puntos de Autoservicio. Esta cantidad asciende a: **238.920 €**.

Por otra parte, los datos utilizados para calcular el coste económico del Proyecto, permite estimar las horas dedicadas a la realización del mismo. Así pues, las cifras anteriores permiten constatar que se han empleado 602 horas para el desarrollo de las fases de Inicio, Elaboración 1 y Elaboración 2 y la implementación física de la Base de Datos. No obstante, a esta cifra hay que añadirle la dedicación realizada para el desarrollo de esta documentación, dedicación cifrada en 90 horas. Así pues, se puede afirmar que se han dedicado un total de **692 horas** para el desarrollo de este Proyecto Final de Carrera.

12 Conclusiones

Una vez concluido el proyecto de especificación y diseño de los Puntos de Autoservicio del SOC es el momento de analizar y valorar el proceso de realización de dicho proyecto, así como los objetivos culminados.

En primer lugar, cabe destacar que el proyecto me ha permitido aplicar y profundizar en contenidos de muy diversa tipología explicados a lo largo de la carrera en las asignaturas de Ingeniería del Software. La diversidad de estos contenidos viene determinada por el mero hecho de que este proyecto se ha desarrollado **desde cero**, ya que, partiendo de un documento de especificaciones técnicas proporcionado por el cliente, se ha procedido a realizar todo el proceso de especificación y diseño del Sistema. Esta realización del Sistema desde cero ha permitido, por tanto, adentrarse en cada una de las etapas del proceso de desarrollo de un sistema *software*, desde la definición y evaluación de los requisitos del Sistema hasta el diseño e implementación de la Base de Datos del mismo.

Otra de las características del proyecto que han permitido que el desarrollo del mismo fuese realmente interesante y enriquecedor es el hecho de que se trate de un **proyecto real**. A pesar de que a lo largo de las diversas asignaturas de Ingeniería del Software se intenta formar a los alumnos para afrontar y solventar las etapas de desarrollo de un sistema, esta formación se limita a simulaciones de problemas reales que tienen un carácter educacional. En cambio, este Proyecto de Final de Carrera se ha desarrollado en base a unas necesidades reales de un cliente que requiere de un sistema *software* que las solventa. Esta característica ha sido precisamente la que ha permitido profundizar enormemente en los contenidos impartidos a lo largo de la carrera.

Además, existen una serie de características intrínsecas a los Puntos de Autoservicio que confieren a este proyecto un carácter especial. En primer lugar, cabe destacar especialmente el hecho de que se trate de una serie de terminales de autoservicio. El desarrollo de este tipo de sistemas no es especialmente frecuente y sus características están bastante alejadas de otros sistemas más habituales como los orientados a medios *Web*. Por otra parte, como se ha comentado a lo largo esta documentación, se trata de un Sistema en el que la interficie gráfica del mismo juega un papel fundamental en las posibilidades de éxito del proyecto. Esta característica obliga a destinar gran parte del esfuerzo realizado a un correcto y adecuado diseño de dicha interficie, esfuerzo que en la mayoría de sistemas se dedica al diseño de las funcionalidades ofrecidas. Esta serie de aspectos han obligado a adaptar y ajustar ciertas metodologías impartidas en las asignaturas de Ingeniería del Software.

Por otra parte, el desarrollo de este proyecto me ha ayudado a estimar la enorme importancia que puede tener en el diseño de un sistema las **características de los usuarios** del mismo, ya que, como se ha podido observar a lo largo de esta documentación, en cada una de las decisiones importantes tomadas, se ha valorado notablemente la inexperiencia de los usuarios en su interacción con sistemas similares.

Por lo que respecta a la culminación de los objetivos marcados al inicio del proyecto, hay que destacar que tal y como se propuso, este proyecto no se ha limitado a proporcionar un marco de diseño sobre el que realizar la implementación física de

los Puntos de Autoservicio, sino que se han realizado una serie de tareas asociadas a la implementación del Sistema tales como la **creación de la interfaz gráfica del mismo** o la **definición de la Base de Datos**. Por tanto, tras la realización de este proyecto, se puede constatar que la finalización del proceso de creación de los Puntos de Autoservicio del SOC se limita únicamente a la implementación de la Capa de Dominio del Sistema, implementación que se sustentará en el diseño realizado en este proyecto.

Así pues, una vez finalizado este proyecto, se puede afirmar que se han cumplido cada uno de los objetivos marcados al inicio del proyecto, ya que se dispone de:

- Guía del proyecto realizada en base a una **instanciación de la metodología RUP** con el objetivo de adaptarla al proyecto de Puntos de Autoservicio.
- Definición del modelo de negocio del *Servei d'Ocupació de Catalunya* con el fin de facilitar la comprensión de las necesidades que el sistema que se va a desarrollar debe satisfacer.
- Definición de los requisitos funcionales y no funcionales del Sistema, prestando especial atención a estos últimos.
- Definición de una planificación y gestión detallada de los recursos facilitando de este modo el desarrollo del proyecto.
- Análisis y diseño de cada una de las funcionalidades descritas en el documento de preinscripciones técnicas aportado por el cliente.
- Diseño e implementación de una interfaz gráfica del Sistema adaptada a las necesidades de los usuarios.
- Diseño e implementación de la Base de Datos del Sistema en el SGBD PostgreSQL.

El Sistema de Puntos de Autoservicio diseñado permite abrir una **nueva vía de comunicación** entre los ciudadanos de a pie y el SOC. Esta nueva vía agiliza y optimiza la realización de una serie de trámites burocráticos, facilitando considerablemente dicha comunicación. Para garantizar el éxito de estas facilidades, las necesidades y características de los ciudadanos el eje fundamental de este proyecto.

Por otra parte, se espera que en un **futuro** cercano el Sistema de Puntos de Autoservicio sea **accesible vía Web**. No obstante, considero que esta ampliación no debería limitarse a la simple incorporación de un nuevo medio de acceso al Sistema, ya que la apertura al mundo *Web* amplía enormemente el abanico de posibilidades del Sistema. Considero que esta apertura debería conllevar una ampliación de las funcionalidades del Sistema, ampliación que no debería centrarse únicamente en aumentar el número de servicios que los terminales ofrecen a los ciudadanos, sino que el acceso vía *Web* al Sistema posibilita el acceso al mismo de otros actores que hasta ahora tenían un papel secundario. Me estoy refiriendo concretamente a los ofertantes de Cursos de Formación y Ofertas de Empleo, ya que un acceso *Web* al Sistema de Puntos de Autoservicio permitiría que dichos ofertantes publicasen nuevos Cursos de Formación y Ofertas de Empleo.

Cambiando de tercio, cabe destacar la herramienta principal utilizada para el desarrollo de este proyecto: **Rational Software Modeller 7.0**. Esta herramienta de IBM es quizás la más potente del mercado en cuanto al modelado de UML y ha facilitado, por tanto, el desarrollo de este proyecto. No obstante, cabe remarcar una serie de **problemas o anomalías** que presenta dicha herramienta en la definición de



los Diagramas de Secuencia. Estas anomalías se producen al incluir en un diagrama estructuras condicionales o iterativas, ya que el programa tiene verdaderas dificultades para detectar la finalización de estas estructuras. Estas dificultades se manifiestan al intentar añadir una nueva interacción entre las entidades que conforman un diagrama inmediatamente después de haber cerrado una estructura condicional o iterativa, ya que al incluir esta nueva interacción justo a continuación de dicha estructura, el programa incluye automáticamente la interacción en el interior de la estructura anterior. Se espera, por tanto, que posteriores versiones de esta herramienta solucionen este error, ya que dificulta y ralentiza considerablemente la definición de Diagramas de Secuencia.

Para concluir, destacar que este Proyecto de Final de Carrera ha sido la culminación de un proceso de aprendizaje que se inició en el año 2002. A lo largo de este proceso, he podido descubrir una pequeña muestra de los secretos que esconde el mundo de la Informática, un mundo por el que he sentido fascinación desde que a los 4 años me regalaron mi primer ordenador. Así pues, espero y deseo que la carrera profesional que inicio ahora, me permita continuar explorando los misterios que encierra la Informática en general y la Ingeniería del Software en particular, especialidad hacia la que pretendo enfocar mi futuro profesional.

13 Bibliografía

- [1] **González Blanco R., Botella A.** *Modelat de Negoci: Descrivint el context del sistema.*
- [2] **James & Suzanne Robertson (2009)** *Volere Requirements Specification Template*, consultado en <http://www.volere.co.uk>
- [3] **Costal D., Franch X., Ribera Sancho M., Teniente E. (2005)** *Enginyeria del Software: Especificació. Especificació de sistemes orientats a objectes amb notació UML.*
- [4] **Gómez C., Mayol E., Olivé A., Teniente E. (2003)** *Diseño de sistemas software en UML.*
- [5] **Gamma E., Helm R., Jonhson R., Vlissides J. (2000)** *Design Patterns: Elements of Reusable Object-Oriented Software*
- [6] **The PostgreSQL Global Development Group (2008)** *PostgreSQL 8.3.6 Documentation.*
- [7] **Oracle (2006)** *Oracle Application Development Framework*
- [8] **OWASP (2005)** *Una Guía para Construir Aplicaciones y Servicios Web Seguros*

Webs de Interés

- [9] [http://www.oficinatreball.cat/socweb/opencms/socweb es/home.html](http://www.oficinatreball.cat/socweb/opencms/socweb_es/home.html): Página Web del Servei d'Ocupació de Catalunya.
- [10] <http://www.sleekandsexy.net/principios-heuristicos-usabilidad/>: Descripción de los principios heurísticos de usabilidad definidos por Jakob Nielsen.
- [11] [http://www.iso.org/iso/support/faqs/faqs_widely_used_standards/widely used standards other/information security.htm](http://www.iso.org/iso/support/faqs/faqs_widely_used_standards/widely_used_standards_other/information_security.htm): Norma ISO 17799 de seguridad.
- [12] <http://www.akamai.com>: Empresa encargada de realizar las réplicas de los servidores del Sistema.
- [13] <http://www.springsource.org>: Página oficial de la comunidad de desarrollo del Framework Spring.
- [14] <http://www.elholgazan.com/2007/10/comparacin-entre-spring-y-ejb-30.html>: Comparativa entre el Framework Spring y EJB.
- [15] <http://www.programania.net/disenio-de-software/spring-vs-ejb/>: Comparativa entre el Framework Spring y EJB.

[16] [http://www.javahispano.org/contenidos/es/spring es mas demandado que ejb en las ofertas de trabajo/?menuId=NEWS&onlypath=true](http://www.javahispano.org/contenidos/es/spring%20es%20mas%20demandado%20que%20ejb%20en%20las%20ofertas%20de%20trabajo/?menuId=NEWS&onlypath=true): Información sobre el *Framework Spring*.

[17] <http://www.apache.org>: Página oficial de la comunidad de desarrollo del Servidor *Apache*.

[18] <http://mery-multi.blogspot.com/2006/12/servidores-web.html>: Comparativa entre los diferentes Servidores *Web* del mercado

[19] <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=363>: Comparativa entre los diferentes Servidores *Web* del mercado

[20] http://dcdbappl1.cern.ch:8080/dcdb/archive/ttraczyk/db_compare/db_compare.html: Comparación de las características de los SGBDs Oracle, MySQL y PostgreSQL.

[21] <http://www.degesys.com/docs/008/oracle.pdf>: Comparación de las características de los SGBDs Oracle, MySQL y PostgreSQL.

[22] <http://bytezero.org/repo/doc/dbs.pdf>: Comparación de las características de los SGBDs Oracle, MySQL y PostgreSQL.

[23] <http://www.scribd.com/doc/102837/Triggers-en-PostgreSQL>: Descripción del proceso de creación de *triggers* en PostgreSQL.

[24] <http://www.hibernate.org> : Página *web* oficial del *Datamapper Hibernate*

[25] <http://www.developersbook.com/articles/hibernate/Hibernate-Vs-iBatis.php>: Comparativa entre *Hibernate* y *iBatis*.

[26] <http://dbmonster.kernelpanic.pl/>: Descripción de la herramienta *DBMonster* utilizada en las pruebas del Sistema.

[27] <http://jakarta.apache.org/jmeter/>: Información sobre la herramienta *JMeter* utilizada en las pruebas del Sistema.

[28] <http://www.wireshark.org>: Página oficial de la herramienta *Wireshark*.

[29] http://es.wikipedia.org/wiki/Ataque_de_fuerza_bruta: Descripción de los ataques de fuerza bruta.

[30] <http://www.bujarra.com/Brutus.html>: Descripción de la herramienta de fuerza bruta *Brutus*.

[31] <http://tomcat.apache.org/>: Página oficial del servidor de aplicaciones *Apache Tomcat*.

